2

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

A RING MODEL FOR
LOCAL/MOBILE RADIO COMMUNICATIONS
WITH CORRECT PACKET CAPTURE

by

Kim J. Tran

March 1991

Thesis Advisor: Tri T. Ha

Approved for public release; distribution is unlimited

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution is unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Naval Postgraduate School

6b. OFFICE SYMBOL (If applicable)
EC

7a. NAME OF MONITORING ORGANIZATION
Naval Postgraduate School

6c. ADDRESS (City, State, and ZIP Code)
Monterey, CA 93943-5000

7b. ADDRESS (City, State, and ZIP Code)
Monterey, CA 93943-5000

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL (If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO.

11 TITLE (Include Security Classification)
A RING MODEL FOR LOCAL/MOBILE COMMUNICATIONS WITH CORRECT PACKET CAPTURE

12. PERSONAL AUTHOR(S)
TRAN, Kim J.

13a. TYPE OF REPORT
Master's Thesis

13b TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT (Year, Month, Day)
1991 March

15 PAGE COUNT
81

16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Aloha networks; packet capture, channel throughput

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Researchers and scientists have been studying Aloha networks for many years hoping to improve the channel throughput and overcome the inherent instability. In this study, we are going to describe some of the Aloha network features. An application of the slotted Aloha network is considered for local/mobile radio communications. The near/far effect, Rayleigh fading, and shadow fading arise in local/mobile radio communications due to differences in distances and topography between each of the users and the base station. These various effects cause a packet capture effect which improves overall channel throughput but leads to different packet delay times for the various users. An analysis of a ring model for local/mobile radio communications with correct packet capture is considered in this paper. The correct packet capture effect of one-ring and three-ring networks are studied.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Ha, Tri T.

22b. TELEPHONE (Include Area Code)
(408) 646-2788

22c OFFICE SYMBOL
EC/Ha

DD Form 1473, JUN 86 Previous editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
S/N 0102-LF-014-6603 UNCLASSIFIED

Block 19 (Cont'd)

A Markov model is developed for a slotted Aloha network with capture. It is shown that the throughput in such a network is markedly greater than the standard 1/e. Perhaps even more important is the result that such networks are more stable under overload.

A Ring Model for Local/Mobile Radio Communications with Correct Packet Capture

by

Kim J. Tran
Civilian, U.S. Department of Navy
B.S., California State University, Long Beach, 1988

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1991

Author: Kim Tran
Kim J. Tran

Approved by: Tri T. Ha
Tri T. Ha, Thesis Advisor

R. Clark Robertson
R. Clark Robertson, Thesis Co-Advisor

Michael A. Morgan
Michael A. Morgan, Chairman
Department of Electrical and Computer Engineering

ABSTRACT

Researchers and scientists have been studying Aloha networks for many years hoping to improve the channel throughput and overcome the inherent instability. In this study, we are going to describe some of the Aloha network features. An application of the slotted Aloha network is considered for local/mobile radio communications. The near/far effect, Rayleigh fading, and shadow fading arise in local/mobile radio communications due to differences in distances and topography between each of the users and the base station. These various effects cause a packet capture effect which improves overall channel throughput but leads to different packet delay times for the various users. An analysis of a ring model for local/mobile radio communications with correct packet capture is considered in this paper. The correct packet capture effect of one-ring and three-ring networks are studied.

A Markov model is developed for a slotted Aloha network with capture. It is shown that the throughput in such a network is markedly greater than the standard 1/e. Perhaps even more important is the result that such networks are more stable under overload.

TABLE OF CONTENTS

I. INTRODUCTION 1

A. ALOHA NETWORKS 1

1. Pure Aloha 1

2. Slotted Aloha 1

B. ALOHA NETWORK IN LOCAL/MOBILE COMMUNICATIONS 4

C. PACKET CAPTURE EFFECT 4

1. Advantages 4

2. Limitations 4

D. PERFORMANCE MEASUREMENT 5

E. ORGANIZATION OF THE THESIS 5

II. PACKET CAPTURE PROBABILITY OF A RING MODEL 6

A. SYSTEM MODEL 6

B. BPSK BIT ERROR PROBABILITY 10

C. CODING FOR PACKET CAPTURE 14

III. NUMERICAL RESULTS 18

A. A ONE-RING NETWORK 19

B. A THREE-RING NETWORK 20

1. Ring one 20

2. Ring two 21

3. Ring Three 22

IV. SYSTEM THROUGHPUT FOR A THREE RING ALOHA NETWORK 28

V. CONCLUSIONS 34

APPENDIX A 36

APPENDIX B 49

APPENDIX C 66

REFERENCES 69

INITIAL DISTRIBUTION LIST 70

LIST OF TABLES

Table I. THE COEFFICIENTS OF a_d AND d 19

Table II. SIMULATION RESULTS FOR THE PROBABILITY OF CORRECT PACKET CAPTURE q(n) FOR ONE-RING NETWORK 20

Table III. SIMULATION RESULTS FOR THE PROBABILITY OF CORRECT PACKET CAPTURE FOR RING ONE 21

Table IV. SIMULATION RESULTS FOR THE PROBABILITY OF CORRECT PACKET CAPTURE FOR RING TWO 22

Table V. SIMULATION RESULTS FOR THE PROBABILITY OF CORRECT PACKET CAPTURE FOR RING THREE 23

Table VI. SIMULATION RESULTS FOR THE PROBABILITY OF CORRECT PACKET CAPTURE FOR ONE-RING NETWORK 52

Table VII. SIMULATION RESULTS FOR THE PROBABILITY OF CORRECT PACKET CAPTURE FOR RING ONE 55

Table VIII. SIMULATION RESULTS FOR THE PROBABILITY OF CORRECT PACKET CAPTURE FOR RING TWO 58

Table IX. SIMULATION RESULTS FOR THE PROBABILITY OF

Table IX. SIMULATION RESULTS FOR THE PROBABILITY OF CORRECT PACKET CAPTURE FOR RING THREE 61

LIST OF FIGURES

Figure 1. The Collision in Pure ALOHA Network. 2

Figure 2. The Collision in Slotted ALOHA Network. 3

Figure 3. System Model 7

Figure 4. User Traffic Density. 9

Figure 5. Probability Of Correct Packet Capture For One-Ring Network 24

Figure 6. Probability Of Correct Packet Capture For Ring One 25

Figure 7. Probability Of Correct Packet Capture For Ring Two 26

Figure 8. Probability Of Correct Packet Capture For Ring Three 27

Figure 9. The Possible State Transition For Each Individual Terminal . 30

Figure 10. System Throughput For A Three-Ring Network 33

Figure 11. Probability Of Correct Packet Capture For One-Ring Network 62

Figure 12. Probability Of Correct Packet Capture For Ring One 63

Figure 13. Probability Of Correct Packet Capture For Ring Two 64

Figure 14. Probability Of Correct Packet Capture For Ring Three 65

I. INTRODUCTION

An analysis of the performance of a local/mobile radio communications system utilizing the slotted ALOHA random access protocol is investigated in this thesis.

A. ALOHA NETWORKS

1. Pure Aloha

In Aloha networks, the information comes in packets. In a conventional pure ALOHA network the users transmit their packets whenever new data are ready for transmission. When two or more users transmit their packets simultaneously, a collision will occur in the channel, and the packets may overlap partially or completely (Figure 1). The collision destroys the information in the packets, and the affected users have to retransmit their packets. These collisions cause a major degradation in performance, which limits the maximum channel throughput of pure ALOHA to 1/2e or 0.184 [Ref. 1].

2. Slotted Aloha

In a slotted ALOHA network, time is segmented into slots. Each slot has a duration equal to the packet transmission time or length. Users are required to synchronize the transmission of their packets so that the leading edges of the packets are aligned with the beginning of the predetermined time slot at the receiver

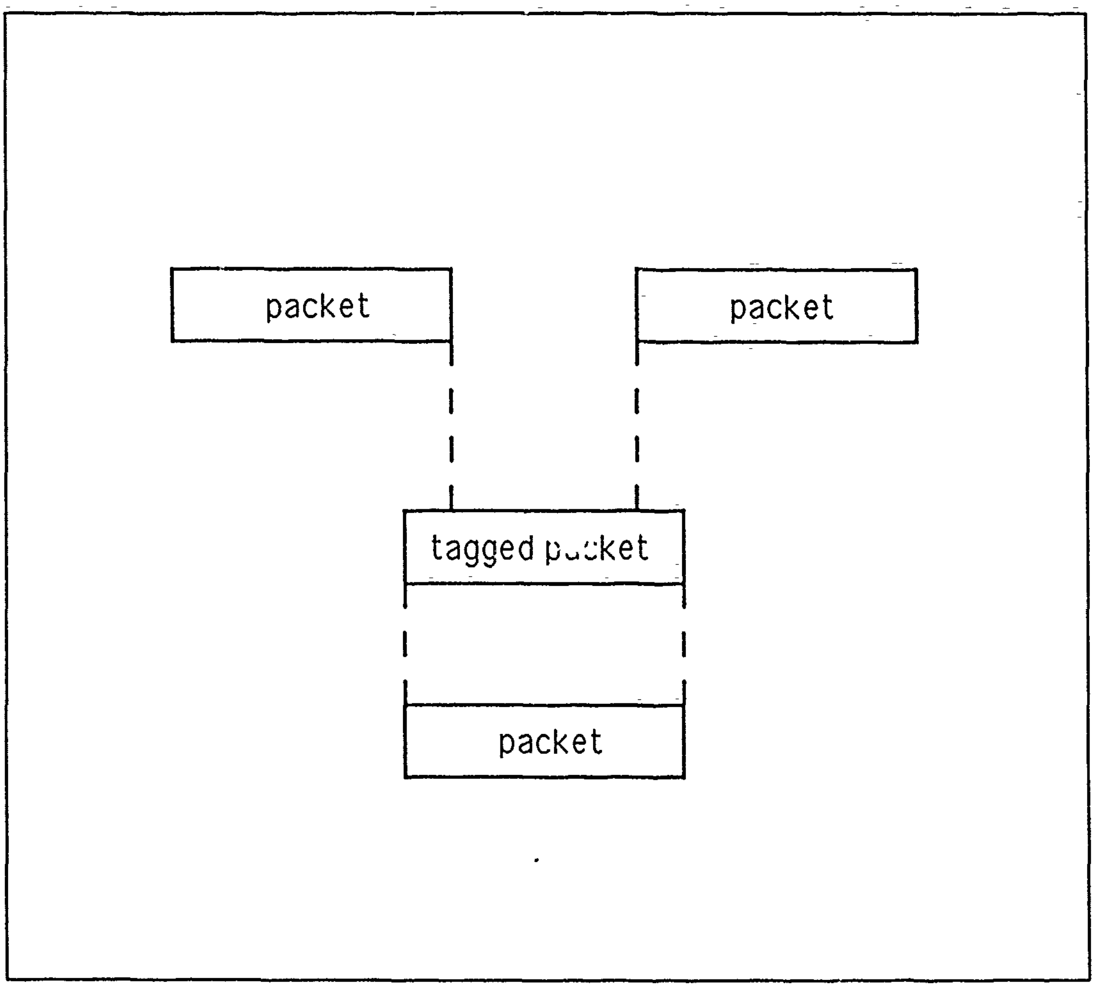

Figure 1. The Collision in Pure ALOHA Network.

Since packets arrival at the receiver are synchronized, when more than one packet is transmitted simultaneously, a collision will occur in the channel, and the packets will overlap completely (Figure 2). The elimination of partial packet overlap substantially improves the performance of the network. The maximum channel throughput is increased to 1/e or 0.368 (twice the maximum channel throughput of pure Aloha) [Ref. 1]. This maximum channel throughput is not necessarily the

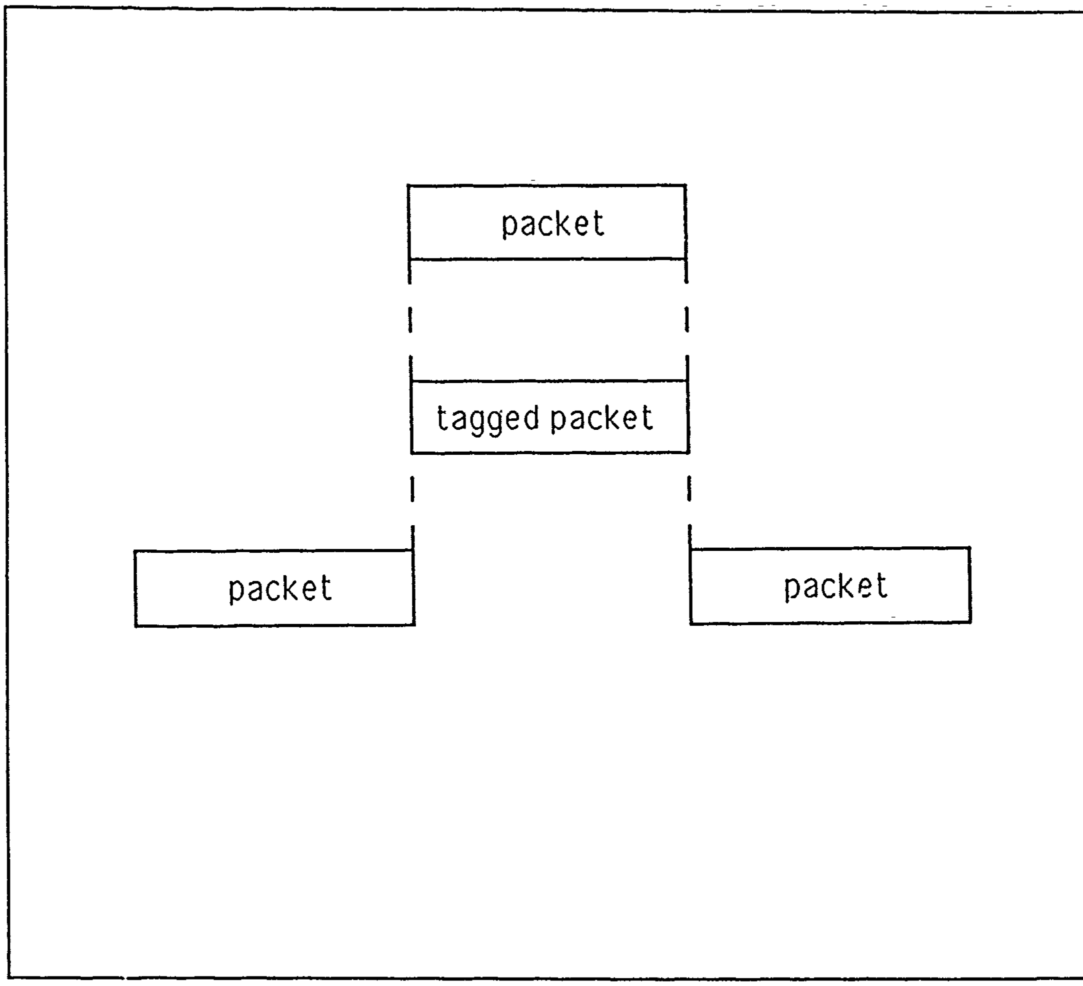

Figure 2. The Collision in Slotted ALOHA Network.

throughput attained by the system. In heavy traffic, the likelihood of collisions increases significantly. After each collision, the affected users have to retransmit their collided packets. As the number of retransmitted packets becomes large, the channel will reach a point where a collision occurs every time slot. Hence, no packets can be transmitted successfully and the system is said to be unstable.

B. ALOHA NETWORK IN LOCAL/MOBILE COMMUNICATIONS

The Aloha network is receiving a great deal of interest for use in applications which involve large physical areas (eg. ships, multi-story buildings, warehouses, urban environments, etc.) and require random multiple access. The two primary limitations of the Aloha network are its poor maximum throughput and its inherent instability. However, the Aloha protocol is still desirable because it is easy to implement.

C. PACKET CAPTURE EFFECT

1. Advantages

In local/mobile radio communications, each transmitter produces a different signal power level at the receiver due to the differences in receiver-transmitter range, shadow fading, and Rayleigh fading. The arriving packet with the highest power level will have the best chance to capture the receiver. As a result, it is possible for the receiver to accurately decode one of the packets involved in a collision. Since the collision does not necessarily destroy all involved packets, the channel throughput will increase considerably. The capture effect significantly improves the performance of the Aloha network [Ref. [illegible]].

2. Limitations

The advantage of improving channel throughput is achieved in favor of users near the base station and at the expense of users far from the base station. The user far from the base station will have a low probability of capturing the base station receiver, and consequently will have to transmit its packet more often than a user near the base station and hence suffer a long packet delay [Ref. 2].

D. PERFORMANCE MEASUREMENT

Several methods have been considered to measure the performance of the Aloha system. They are packet reception probability, the correct reception of the packet address header, and capture measures based on a threshold. Since the packet reception probability provides the best measure, this method is selected to measure the performance of the system. A transmitted packet is considered to have captured the base station receiver only when the entire packet is received correctly in the presence of one or more interfering packets [Ref. 3].

E. ORGANIZATION OF THE THESIS

The analysis is divided into four sections. Section II investigates the capture effect for a ring model in local/mobile communications. Section III presents the numerical results for a three-ring network. Section IV analyzes the system throughput for a multiple ring Aloha network using a Markov chain model. Finally, discussions and conclusions are presented in Section V.

II. PACKET CAPTURE PROBABILITY OF A RING MODEL

The analysis of the packet capture effect in a ring model for local /mobile radio systems will be presented in this section. The probability of correct packet capture will be evaluated for a system using ideal coherent binary phase-shift-keying (BPSK) modulation.

A. SYSTEM MODEL

A network consisting of uniformly distributed users in an annular ring will be studied. A uniformly distributed density function for the user traffic is assumed. The model illustrated in Figure 3 is the system model for this study.

The traffic generated within a distance r of the base station is

$$G(r) = 2\pi \int_0^r x g(x)\, dx \tag{1}$$

where r is the distance from the user to the base station. The user traffic density is g(r).

The total offered traffic in the network is obtained as $G = G(r \to \infty)$. Therefore

$$G = 2\pi \int_0^\infty x g(x) dx \tag{2}$$

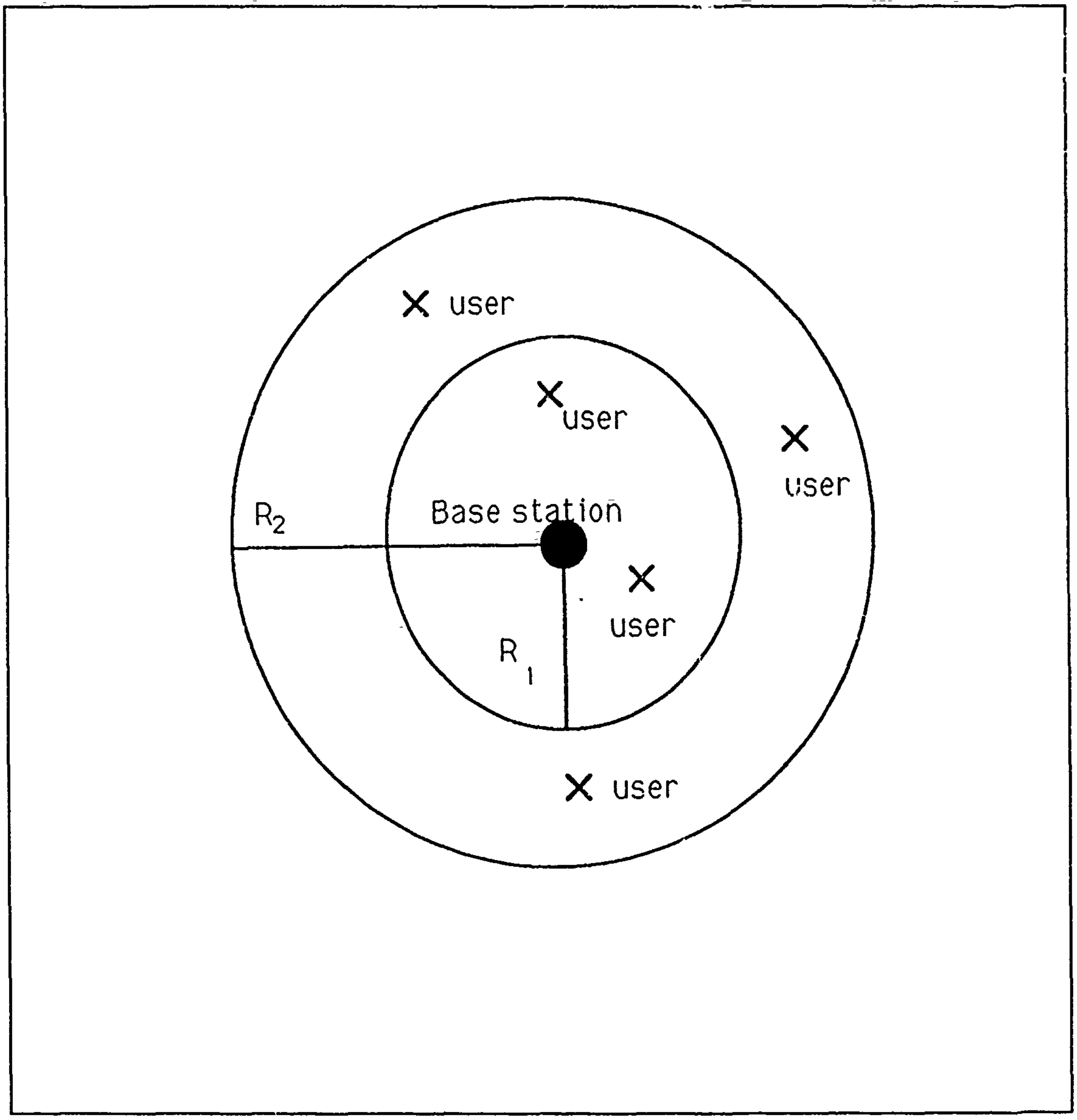

Figure 3. System Model

The spatial distribution function for packet generation (the probability of a a packet generated within a distance $R \leq r$) is [Ref. 3]

$$F_R(r) = Pr\{R \le r\} = \frac{2\pi}{G}\int_0^r x g(x) dx \tag{3}$$

Therefore the probability density function for packet generation is found to be

$$f_R(r) = \frac{dF_R(r)}{dr} = \frac{2\pi}{G} r g(r) \tag{4}$$

The probability density function for packet generation in ring R_i, defined as $r_i \le r \le r_{i+1}$, is

$$f_R(r|R_i) = \begin{cases} \dfrac{f_R(r)}{F_R(r_{i+1}) - F_R(r_i)}, & r_i \le r \le r_{i+1} \\ 0, & otherwise. \end{cases} \tag{5}$$

Assuming that users are uniformly distributed in an annular ring of inner radius R_1 and outer radius R_2, we have the user traffic density

$$g(r) = \begin{cases} \dfrac{G}{\pi(R_2^2 - R_1^2)}, & R_1 \le r \le R_2 \\ 0, & otherwise \end{cases} \tag{6}$$

Substituting (6) into (4), we have the probability density function for packet generation as follows

$$f_R(r) = \begin{cases} \dfrac{2r}{R_2^2 - R_1^2}, & R_1 \le r \le R_2 \\ 0, & otherwise \end{cases} \tag{7}$$

The distribution of packet generation of a ring with $R_1 \le r \le R_2$ is shown in Figure 4. The probability of a packet being generated within a ring of radius $R < r_i$ is given by

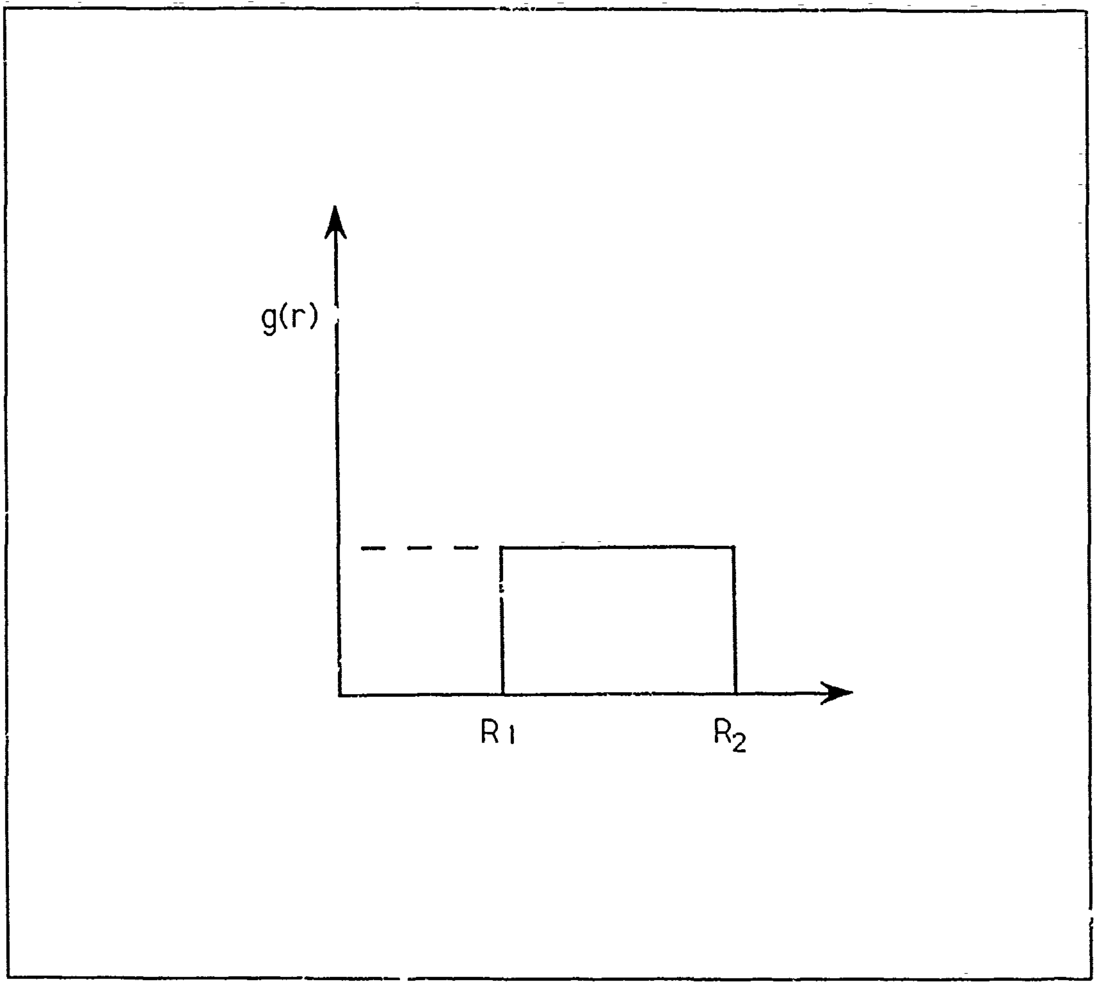

Figure 4. User Traffic Density.

$$F_R(r_i) = \frac{2\pi}{G}\int_0^{r_i} x \frac{G}{\pi(R_2^2 - R_1^2)} dx \tag{8}$$

which can be evaluated to yield

$$F_R(r_i) = \frac{r_i^2}{R_2^2 - R_1^2} \tag{9}$$

The probability of a packet generated within a ring of radius $R < r_{i+1}$ is

$$F_R(r_{i+1}) = \frac{r_{i+1}^2}{R_2^2 - R_1^2} \tag{10}$$

Substituting (10) into (5) and (7), we get the probability density function for packet generation in ring R_i

$$f_R(r|R_i) = \frac{2r}{r_{i+1}^2 - r_i^2}, \quad r_i \le r \le r_{i+1} \tag{11}$$

where i equals 0, 1, 2,...,n-1 and $R_i = \{r_i \le r \le r_{i+1}\}$

B. BPSK BIT ERROR PROBABILITY

We assume that users communicate with the base station using coherent binary phase shift keying (BPSK). For ideal coherent binary phase shift keying (BPSK) modulation in a Rayleigh fading channel, the received signal of a particular user 0 in the presence of n-1 interfering users is given by (assuming the background Gaussian noise is much smaller than the interference)

$$r_0(t) = c_o s_o(t) + \sum_{i=1}^{n-1} c_i \, e^{-j\theta_i} s_i(t) \tag{12}$$

where $s_i(t)$ are BPSK signals, the c_i's are Rayleigh distributed signal amplitudes, and the θ_i's are the uniformly distributed phases of the interfering signals (we set $\theta_0 =$ 0 and assume that θ_i, i=1, 2,..., n-1 are relative to this reference).

At the sampling time the decision variable of BPSK receiver is

$$v_0 = c_0 a_0 + \sum_{i=1}^{n-1} c_i \, a_i \cos\theta_i \tag{13}$$

where $a_i = \{\pm 1\}$ and are independent and identical distributed random variables (i.i.d.). The second term on the right side of the above expression is the sum of independent Gaussian random variables.

The received signal power is

$$S = E\,\{(c_0\,a_0)^2\} = E\,\{c_0^2\} \tag{14}$$

The received noise power is

$$N = E\,\{(v_0 - c_0\,a_0)^2\} \tag{15}$$

Substituting equation (13) into equation (15), we have the received noise power as follows

$$N = E\,\{(\sum_{i=1}^{n-1} c_i\,a_i \cos\theta_i)^2\} \tag{16}$$

Noting that the mean of $\cos^2\theta_i$ is 1/2, we find

$$N = \frac{1}{2} \sum_{i=1}^{n-1} E\{c_i^2\} \tag{17}$$

Hence, the received signal to noise ratio is

$$SNR = \frac{S}{N} = \frac{2E\{c_0^2\}}{\sum_{i=1}^{n-1} E\{c_i^2\}} \tag{18}$$

The bit error probability of user 0 in the presence of n-1 interfering users for BPSK in a Rayleigh fading channel is [Ref. 4]

$$P_b = \frac{1}{2}\left[1 - \frac{1}{\sqrt{1+\frac{2}{SNR}}}\right] \tag{19}$$

The near/far effect is modeled by assuming that the received power w conforms to an inverse α power propagation law; that is

$$w = cr^{-\alpha} \tag{20}$$

where r is the distance of the user from the base station and α is the propagation constant. Generally, $3 \leq \alpha \leq 5$ for ultra high frequency (UHF) applications. The constant c depends on the gains, heights, and power of the transmitter and receiver antennas. Without loss of generality, we assume that c is normalized to unity for all users; that is, users are taken to be identical and use omnidirectional antennas with radiation pattern maxima in the horizontal plane.

From (20), the received power of user i is

$$w_i = E\{c_i^2\} = r_i^{-\alpha} \tag{21}$$

Substituting (21) into (18), we have

$$SNR = \frac{2w_0}{\sum_{i=1}^{n-1} w_i} = \frac{2r_0^{-\alpha}}{\sum_{i=1}^{n-1} r_i^{-\alpha}} = \frac{2}{\sum_{i=1}^{n-1} \left(\frac{r_i}{r_0}\right)^{-\alpha}} \tag{22}$$

and the bit error probability is

$$P_b = \frac{1}{2}\left[1 - \frac{1}{\sqrt{1+\sum_{i=1}^{n-1}\left(\frac{r_i}{r_0}\right)^{-\alpha}}}\right] \tag{23}$$

which can be written as

$$P_b = \frac{1}{2}\left[1 - \frac{1}{\sqrt{1+\sum_{i=1}^{n-1}\left(\frac{r_0}{r_i}\right)^{\alpha}}}\right] \tag{24}$$

For numerical analysis purposes, we select the propagation constant $\alpha = 4$.

Hence,

$$P_b = \frac{1}{2}\left[1 - \frac{1}{\sqrt{1+\sum_{i=1}^{n-1}\left(\frac{r_0}{r_i}\right)^{4}}}\right] \tag{25}$$

The bit error probability for the case of M-level diversity is [Ref. 3]

$$P_{bM} = P_b^M \sum_{k=0}^{m-1} \binom{M-1+k}{k} \left(\frac{1+\mu}{2}\right)^k \qquad (26)$$

where the value of μ is given by

$$\mu = \left[1+\sum_{i=1}^{n-1}\left(\frac{r_0}{r_i}\right)^4\right]^{-\frac{1}{2}} \qquad (27)$$

C. CODING FOR PACKET CAPTURE

Since all packets are transmitted with equal power and because of the near/far effect and the Rayleigh fading channel, the operating signal-to-noise ratio can be very low. To overcome this problem, we propose a combination of diversity and forward error correction. The Viterbi decoding scheme is employed. This scheme proposes maximum likelihood decoding, which is relatively easy to implement for codes with small memory orders. A binary convolutional code with rate R = 1/2 and constraint length 7, and Viterbi hard decision decoding is considered. Define the input modulation as M-ary and the output modulation as Q-ary. When binary coding is used, the modulator has only binary input (M = 2). Similarly, when binary demodulator output quantization is used (Q = 2), the decoder has only binary output. In this case, the demodulator is said to make hard decisions (M = Q). To simplify the implementation, most coded digital communication systems use binary coding with hard-decision decoding. Since errors at the output of the Viterbi decoder occur in burst even if the errors at the input of the decoder are independent, the probability of correct packet reception cannot be evaluated by assuming independent errors. It has been shown that probability of correct packet reception P_C is lower bounded by [Ref. 3]

$$P_C \geq (1-P_E)^L \tag{28}$$

where L is the packet length and P_E is the union bound first event error probability [Ref. 5]. For hard decision decoding, a tight upper bound on P_E is given by

$$P_E \leq \sum_{d=d_f}^{\infty} a_d P_d \tag{29}$$

where a_d is the number of code words of Hamming weight d, d_f is the free distance of the convolutional code, and P_d is the first event error probability of a code word of weight d. The first event error is made with probability [Ref. 5]

$$P_d = \begin{cases} \displaystyle\sum_{e=(d+1)/2}^{d} \binom{d}{e} P_{bM}^{e} (1-P_{bM})^{d-e} & , d \text{ odd} \\ \displaystyle\frac{1}{2}\binom{d}{d/2} P_{bM}^{d/2} (1-P_{bM})^{d/2} + \sum_{e=1+d/2}^{d} \binom{d}{e} P_{bM}^{e} (1-P_{bM})^{d-e} & , d \text{ even} \end{cases} \tag{30}$$

The bound of (29) can be further simplified by noting that for d odd

$$P_d < \sum_{e=(d+1)/2}^{d} \binom{d}{e} P_{bM}^{d/2} (1-P_{bM})^{d/2} \tag{31}$$

Expressions involving P_{bM} can be moved out of the summation since P_{bM} is independent of the summation index e. As a result, the upper bound for P_d is

$$P_d < P_{bM}^{d/2}\,(1-P_{bM})^{d/2} \sum_{e=(d+1)/2}^{d} \binom{d}{e} \tag{32}$$

Taking the lower limit of e at 0 instead of (d+1)/2, we obtain

$$P_d < P_{bM}^{d/2}\,(1-P_{bM})^{d/2} \sum_{e=0}^{d} \binom{d}{e} \tag{33}$$

The sum can now be evaluated to obtain an upper bound on P_d of

$$P_d < 2^d\, P_{bM}^{d/2}\,(1-P_{bM})^{d/2} \tag{34}$$

Similarly, for d even an upper bound on P_d is found to be

$$P_d < \frac{1}{2}\binom{d}{d/2} P_{bM}^{d/2}\,(1-P_{bM})^{d/2} + \sum_{e=d/2+1}^{d} \binom{d}{e} P_{bM}^{d/2}\,(1-P_{bM})^{d/2} \tag{35}$$

This expression can be simplified in a manner similar to that used to obtain (34) to get an upper for P_d when d is even

$$P_d < 2^d\, P_{bM}^{d/2}\,(1-P_{bM})^{d/2} \tag{36}$$

which is identical to (34).

The correct packet capture probability q(n) is obtain by averaging the probability of correct packet reception over all possible location of n users. Thus

$$q(n) = \int_{R_1}^{R_2}\int_{R_1}^{R_2}\cdots\int_{R_1}^{R_2} P_C\, f_R(r_0)\ldots f_R(r_{n-1})\,dr_0\ldots dr_{n-1} \tag{37}$$

where $f_R(r)$ is given by (7). Now substituting (30) into (42), we obtain a lower bound for the correct packet capture probability as

$$q(n) \geq \int_{R_1}^{R_2} \cdots \int_{R_1}^{R_2} (1-P_E)^L f_r(r_0) \cdots f_R(r_{n-1})\, dr_0 \cdots dr_{n-1} \qquad (38)$$

III. NUMERICAL RESULTS

A numerical analysis has been performed. Programs using Matlab provide numerical results for the above correct packet capture probability for user 0. Eight interfering packets have been used in the analysis. As the number of the interfering packets increases the computation time becomes very lengthy. It took approximately 24 hours to compute q(1), q(2)...q(6). Therefore, the simplified P_d in (36) is used to calculate q(1), q(2),... q(7), q(8). The result for q(n) using the simplified bound of P_d in (36) yields results that are very close to those obtained with (30). For each additional user after eight interferers, the results do not change significantly. The source codes in Appendix A are executed to obtained the results which are listed in Table II, III, IV , and IV. The results and source codes using (30) are listed in Appendix B for reference purposes.

Numerical results are obtained for the specific case of L = 400 bits and α = 4. The levels of diversity considered are M = 1, 2, and 4. A convolutional code of rate 1/2, constraint length 7 is used. The values of a_d and d are listed in Table I. [Ref. 3]

Table I. THE COEFFICIENTS OF a_d AND d

a_d	d
11	10
38	12
193	14
1,331	16
7,275	18
40,406	20

We will study a ring model for the following cases.

A. A ONE-RING NETWORK

A one-ring is used for the network. The user r_0 locates in a range from R_1 = 0 km to R_2 = 50 km, any other users will vary in the same range with user r_0. The probability of correct packet capture for one-ring is

$$q(n) = \int_0^{50}\int_0^{50}\cdots\int_0^{50}(1-P_E)^L f_R(r_0) f_R(r_1)\cdots f_R(r_{n-1})\,dr_0\cdots dr_{n-1} \qquad (39)$$

where

$$f_R(r_j) = \begin{cases} \dfrac{2r_j}{50^2} & , 0 \le r_j \le 50 \\ 0 & , otherwise \end{cases} \qquad (40)$$

where j = 0, 1, 2, 3, ..., n-1. The calculated values for q(n) are listed in Table II.

Table II. SIMULATION RESULTS FOR THE PROBABILITY OF CORRECT PACKET CAPTURE q(n) FOR ONE-RING NETWORK

	q(1)	q(2)	q(3)	q(4)	q(5)	q(6)	q(7)	q(8)
M=1	1	.2049	.1116	.0789	.0587	.0459	.0382	.0334
M=2	1	.3676	.2142	.1508	.1116	.0876	.0741	.0649
M=4	1	.6318	.3964	.2801	.2132	.1750	.1464	.1252

B. A THREE-RING NETWORK

The following is an analysis of a three-ring model. Ring one is {0 km ≤ r ≤ 25 km}; ring two is {25 km ≤ r ≤ 38.5 km} and ring three is {38.5 km ≤ r ≤ 50 km}.

1. Ring one

The probability of correct packet capture for ring one is

$$q(n) = \int_0^{50} \ldots \int_0^{50} \int_0^{25} (1-P_E)^L f_R(r_0|R_1) f_R(r_1) \ldots f_R(r_{n-1}) dr_0 \ldots dr_{n-1} \qquad (41)$$

where

$$f_R(r_0|R_1) = \begin{cases} \dfrac{2r_0}{25^2}, & 0 \le r_0 \le 25 \\ 0, & otherwise \end{cases} \tag{42}$$

$$f_R(r_j) = \begin{cases} \dfrac{2j}{50^2}, & 0 \le r_j \le 50 \\ 0, & otherwise \end{cases} \tag{43}$$

where j = 1, 2, 3 ...n-1. The calculated values of q(n) are listed in Table III.

Table III. SIMULATION RESULTS FOR THE PROBABILITY OF CORRECT PACKET CAPTURE FOR RING ONE

	q(1)	q(2)	q(3)	q(4)	q(5)	q(6)	q(7)	q(8)
M=1	1	.6758	.4412	.3221	.2532	.2080	.1770	.1543
M=2	1	.8579	.6960	.5722	.4801	.4036	.3396	.3141
M=4	1	.9042	.8099	.7197	.6350	.5580	.4901	.4615

2. Ring two

The probability of correct packet capture for ring two is

$$q(n) = \int_0^{50}\cdots\int_0^{50}\int_{25}^{38.5} (1-P_E)^L f_R(r_o|R_2) f_R(r_1)..f_R(r_{n-1}) dr_0..dr_{n-1} \tag{44}$$

where

$$f_R(r_0|R_2) = \begin{cases} \dfrac{2r_0}{38.5^2-25^2}, & 25 \le r_0 \le 38.5 \\ 0, & otherwise \end{cases} \tag{45}$$

$$f_R(r_j) = \begin{cases} \dfrac{2r_j}{50^2}, & 0 \le r_j \le 50 \\ 0 & ,otherwise \end{cases} \tag{46}$$

where j = 1, 2, 3,...n-1. The calculated values of q(n) are listed in Table IV.

Table IV. SIMULATION RESULTS FOR THE PROBABILITY OF CORRECT PACKET CAPTURE FOR RING TWO

	q(1)	q(2)	q(3)	q(4)	q(5)	q(6)	q(7)	q(8)
M=1	1	.0199	0	0	0	0	0	0
M=2	1	.2897	.0606	.0100	.0019	0	0	0
M=4	1	.4731	.2581	.1080	.0446	.0166	.0063	.005

3. Ring Three

The probability of correct packet capture for ring three is

$$q(n) = \int_0^{50} \ldots \int_0^{50} \int_{38.5}^{50} (1-P_E)^L f_R(r_0|R_3) f_R(r_1) \ldots f_R(r_{n-1}) \tag{47}$$

where

$$f_R(r_0|R_3) = \begin{cases} \dfrac{2r_0}{50^2-38.5^2}, & 38.5 \le r_o \le 50 \\ 0 & , otherwise \end{cases} \tag{48}$$

$$f_R(r_j) = \begin{cases} \dfrac{2r_j}{50^2}, & 0 \le r_j \le 50 \\ 0 & , otherwise \end{cases} \tag{49}$$

where j = 0, 1, 2,...n-1. The calculated values of q(n) are listed in Table V.

Table V. SIMULATION RESULTS FOR THE PROBABILITY OF CORRECT PACKET CAPTURE FOR RING THREE

	q(1)	q(2)	q(3)	q(4)	q(5)	q(6)	q(7)	q(8)
M=1	1	0	0	0	0	0	0	0
M=2	1	.0133	0	0	0	0	0	0
M=4	1	.2302	.0210	.0008	0	0	0	0

Plots of q(n) versus n for case 1) and each of the rings for case 2) are shown in Figures 5, 6, 7, and 8.

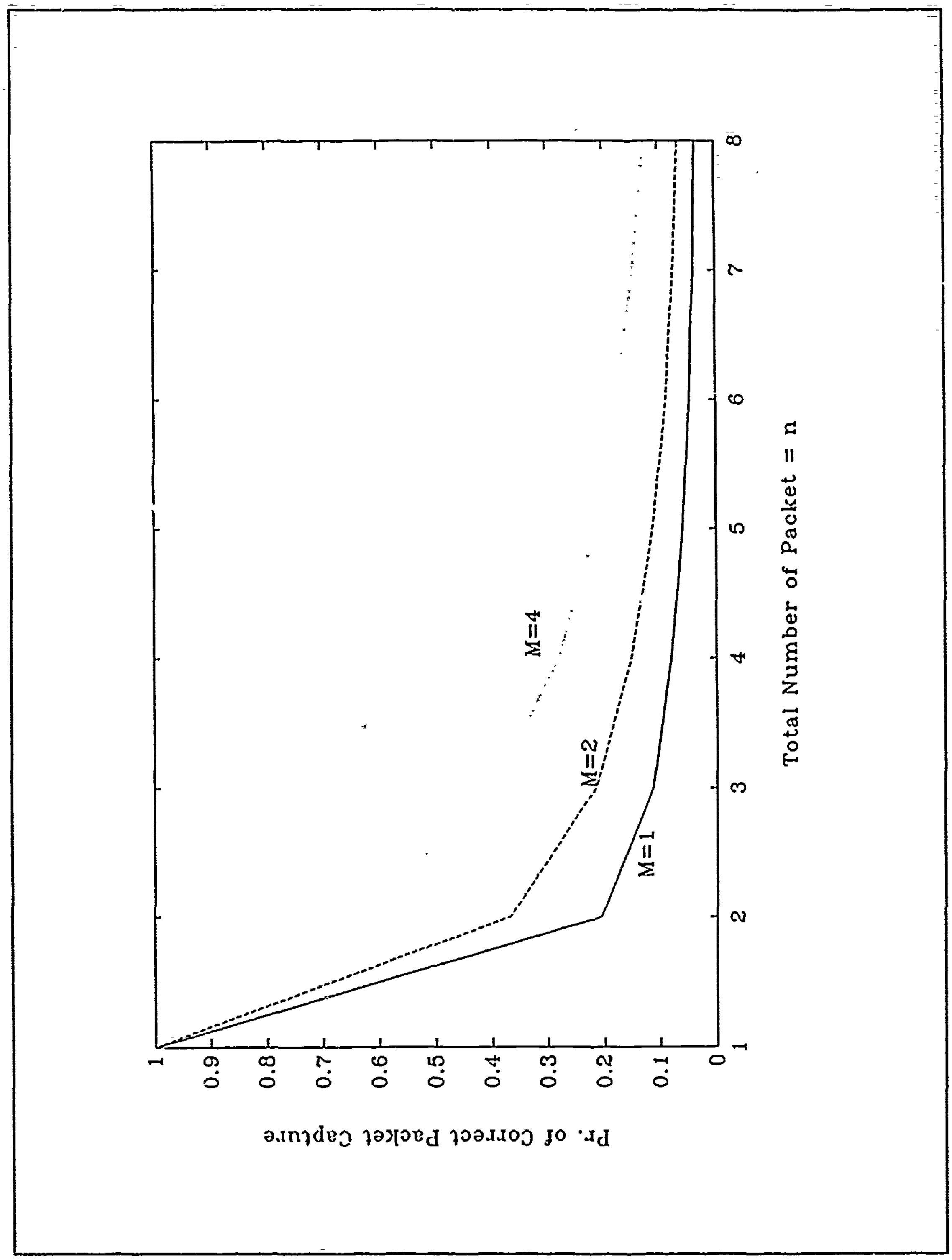

Figure 5. Probability Of Correct Packet Capture For One-Ring Network

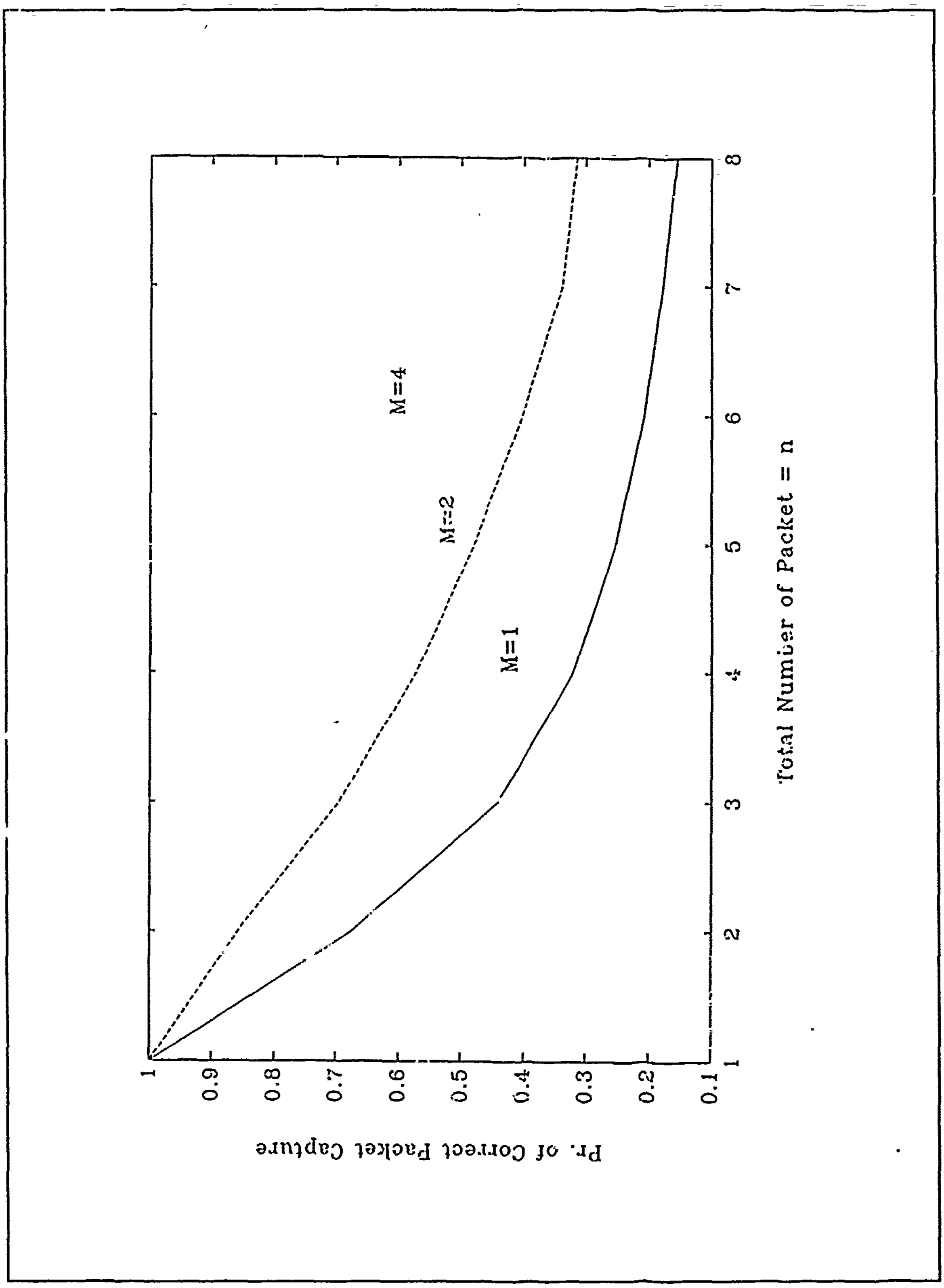

Figure 6. Probability Of Correct Packet Capture For Ring One

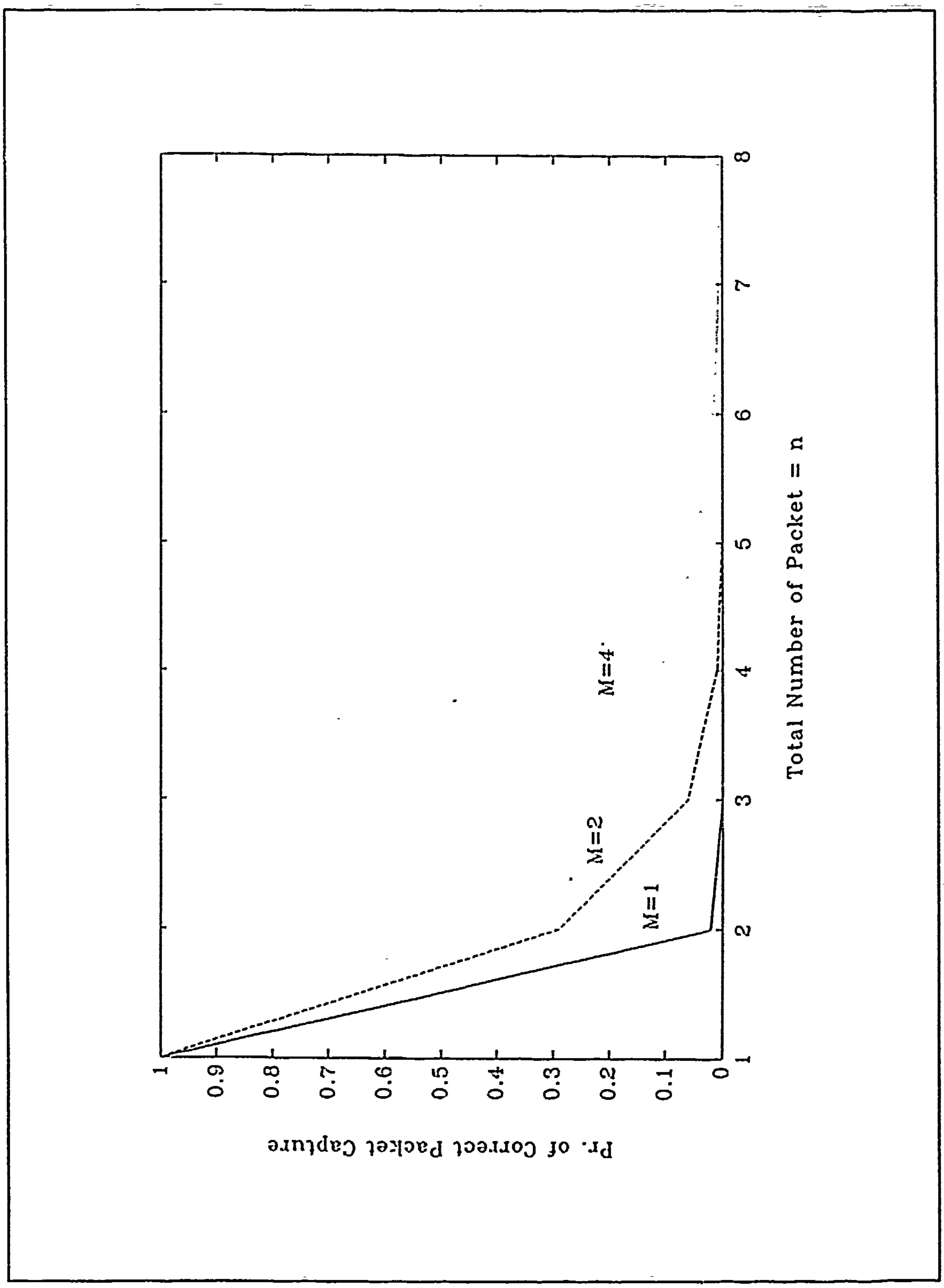

Figure 7. Probability Of Correct Packet Capture For Ring Two

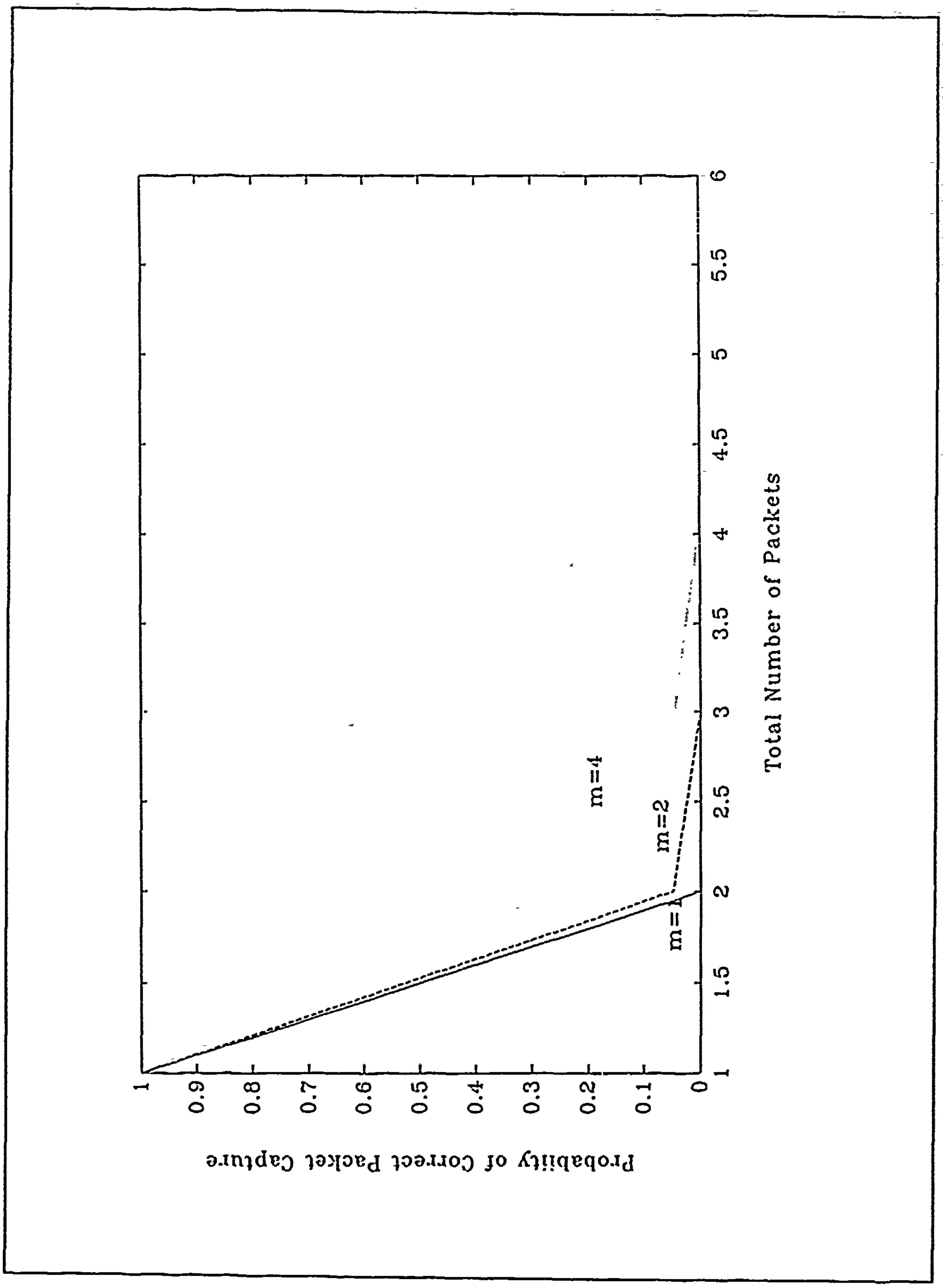

Figure 8. Probability Of Correct Packet Capture For Ring Three

IV. SYSTEM THROUGHPUT FOR A THREE RING ALOHA NETWORK

The analysis of a ring model for mobile/local radio communications with correct packet capture probability has been performed in the previous section. The numerical results show that users who are located close to the base station are more likely to capture the base station. The throughput for each ring will be calculated by using a Markov chain model. The Markov model provides a detailed model of the network's dynamic behavior; therefore, it is a desirable model for the analysis.

The probability q_i that one of i colliding packets will survive a collision and that the other i-1 packets will need retransmission later has been calculated in Section III. The previous system model of an Aloha network with N terminals (users) sharing one channel for transmission to one receiver (base station) is used. It is assumed that positive acknowledgements from the receiver are never lost, so the user will not retransmit unnecessarily. This assumption has been shown to be very realistic since modern mobile networks usually have to guarantee a ninety five percent coverage of the serving area. The values of q_i are calculated in Section III, and it is assumed that q_i does not equal to zero for i greater than one. This assumption also depends on the specific properties of the network. In an undisturbed channel, q_1 is equal to 1 and q_0 is equal to 0 [Ref. 8].

During any time slot each terminal (user) will be in any of three states: the original state O, the transmission state T, or the retransmission state R. The terminal will be in state O when there is no packet ready to transmit or if a new packet has just arrived while it is waiting for the next time slot. The terminal will be in state T when it is actually transmitting a packet, successfully or not. The terminal will be in state R after it has transmitted a packet unsuccessfully or when it is waiting for a future time slot for retransmission. The terminal will be designated as an O-, T-, or R-terminal depending on which state it is in. It is assumed that the process which generates new packets will be halted in a terminal while it has a packet to transmit. The possible state transitions for an individual terminal are shown in Figure 9.

It is assumed that changes of state can occur at the beginning or at the end of the time slot in the Aloha network. An O-terminal can either remain in state O (do nothing) or change into a T-terminal at the beginning of a time slot. Upon changing into a T-terminal, it starts the transmission of a new packet. The generation rate which a terminal changes from state O to state T is denoted p_0. An R-terminal can either stay in state R (do nothing) or change into a T-terminal at the beginning of the time slot. In the latter case, it starts the retransmission. The retransmission rate when a terminal changes from R to T will be denoted p_r. At the end of a time slot, a T-terminal can either go to O after a successful transmission or to R after an unsuccessful transmission.

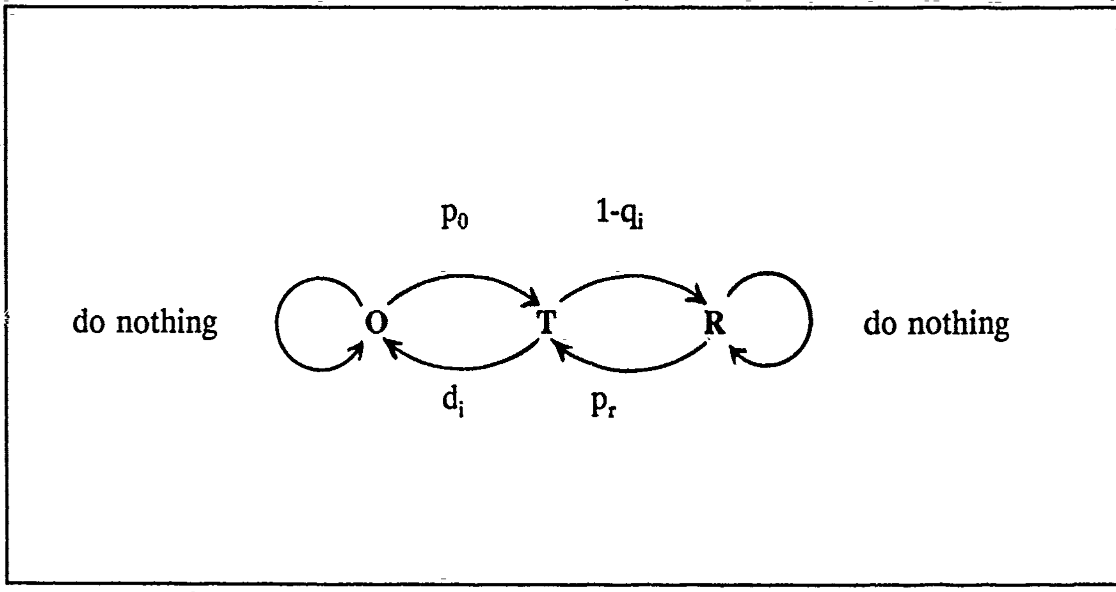

Figure 9. The Possible State Transition For Each Individual Terminal

A slotted Aloha network can be described as a homogenous finite Markov process with N+1 possible states (0, 1, ...N) corresponding to the number of R-terminals at the end of each time slot.

Given $0 \leq n \leq N$ and $0 \leq m \leq N$, let $\pi_{n,m}$ denote the state transition probability that the network will move from state n to m in one step. There are three different types of transitions:

Type 1, in which the state decreases more than one. This is not possible because only one packet will be transmitted successfully in one time slot.

Type 2, in which the state decreases by one with probability q_i. This will happen if and only if there is no O-terminal and i R-terminals change into state T for $1 \leq i \leq n$ (one packet is successful).

Type 3, the state moves to $m \geq n$. This will happen with probability $q_{i+m-n+1}$ if and only if m-n+1 O-terminals and i R-terminals change into state T for $0 \leq i \leq n$; or with probability $1-q_{i+m-n}$ if and only if m-n O-terminals and i R-terminals change into state T for $0 \leq i \leq n$ (no packet successful).

It is assumed that the terminals are operating independently. The generation rate and the retransmission rate are denoted p_0 and p_r, respectively. The probability of correct packet capture, q_i, was obtained in Section III. The value of $\pi_{n,m}$ is calculated as [Ref. 8]

$$\pi_{n,m} = \begin{cases} 0 & , \text{if } m < n-1 \\ (1-p_0)^{N-n} \cdot \sum_{i=1}^{n} \binom{n}{i} \cdot p_r^i \cdot (1-p_r)^{n-1} \cdot q & , \text{if } m = n-1 \\ \left[\binom{N-n}{N-m} \cdot (1-p_0)^{N-m-1} \cdot p_0^{m-n} \cdot \sum_{i=0}^{n} \binom{n}{i} \cdot (1-p_0)^{n-i} \cdot \rho_r \right] & \\ \quad \cdot \left[(1-p_0) \cdot (1-q_{i+m-n}) + \frac{N-m}{m-n+1} \cdot p_0 \cdot q_{i+m-n+1} \right] & , \text{if } m \geq n \end{cases} \tag{50}$$

To calculate the equilibrium state occupation probabilities π_n^*, an arbitrary positive number is assigned to π_n^*. The π_n^* is calculated as follows [Ref. 6]

$$\pi_n^* = \frac{1}{\pi_{n,n-1}} \cdot \left(\pi_{n-1}^* - \sum_{i=0}^{n-1} \pi_i^* \pi_{i,n-1} \right) \tag{51}$$

Normalizing the π_n^*, one gets

$$\pi_n = \frac{\pi_n^*}{\sum_{i=0}^{N} \pi_i^*} \tag{52}$$

The expected throughput of the network in state n (the probability to transmit a packet successfully in a time slot) is found to be

$$f_n = \sum_{k=0}^{N-n} \binom{N-n}{N-n-k} \cdot p_0^k \cdot (1-p_0)^{N-n-k} \cdot \sum_{i=0}^{n} \binom{n}{i} \cdot p_r^i \cdot (1-p_r)^{n-i} \cdot q_{i+k} \tag{53}$$

The throughput for a ring is

$$S = \sum_{n=0}^{N} \pi_n f_n \tag{54}$$

The system throughput is shown in Figure 10. The curves S1, S2, and S3 are the throughputs of ring one, two, and three, respectively. The curve S1+S2+S3 is the overall throughput of the network. The source code for Figure 10 is listed in Appendix C.

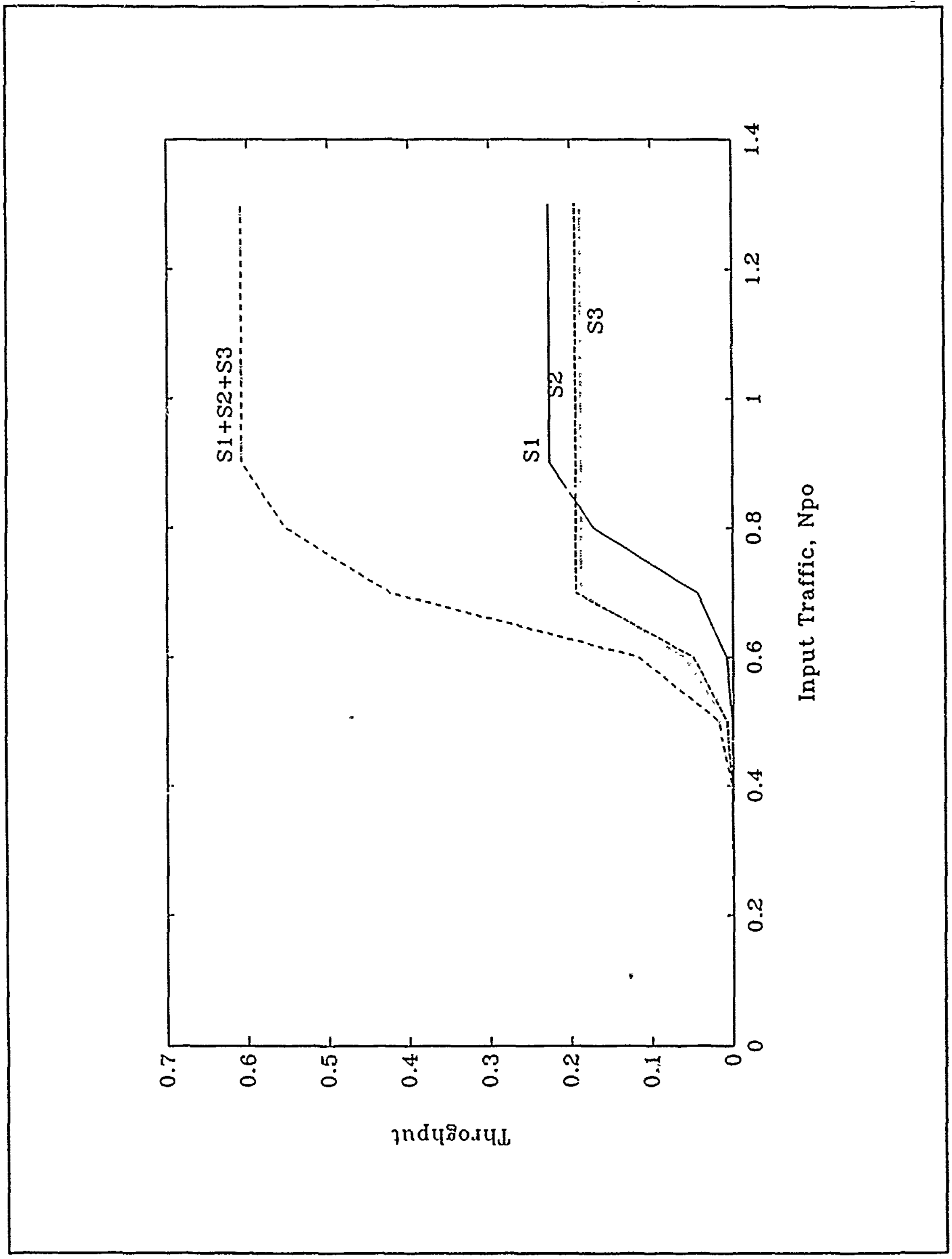

Figure 10. System Throughput For A Three-Ring Network

V. CONCLUSIONS

At first, an introduction to Aloha networks have been made. The slotted Aloha network is selected in this study, since its system throughput is double the pure Aloha system throughput. Then, the probability of correct packet capture of slotted Aloha is studied for mobile/local radio communications. An application of slotted Aloha in a ring model has been developed. In this model, the collisions do not necessarily destroy all the involved packets. With the help of this model, the probability of correct packet capture has been investigated. It has been shown that for a ring network the system throughput is greater than 1/e, and the ring network is more stable under overload. The numerical results in Section III have been plotted in Figures 5, 6, 7, and 8. The probability of correct packet capture of one-ring and three-ring networks have been studied. The numerical results and its plot in Figure 5 show that as the number of users grows from one to six the probability of correct packet capture drops rapidly. For each additional user after six users, the probability of correct packet capture does not decrease as quickly as with only a few users. When the number of users is large, the system reaches a steady state. The probability of correct packet capture increases as M (diversity level) increases.

Numerical results for a three-ring network show that when the users are located in the inner ring (e.g. ring one) they will have a better chance to capture the receiver.

As the users move to the outer ring (eg. ring three), the probability that they will capture the base station decreases rapidly. The system throughputs for each ring have been calculated in Section IV based on the probability of correct packet capture obtained previously.

APPENDIX A

As mentioned in Section III, equation (30) is used to calculate q(n) in Tables II, III, IV , and V. The source codes for the results shown in Tables II, III, IV, and V are listed below. Since q(6), q(7), and q(8) require a long computation time, the program is written to calculate each value of q(n) separately for large value of n (n > 6).

The source code for the numerical results of Table II is listed below. The number of users (nuser) is initialized to 2, 3,..., and 8 to calculate q(2), q(3), ..., q(8), respectively. In the listed program the maximum number of users (nuser) is equal to eight. For a one ring network, R_1 and R_2 are varied from 0 km to 50 km.

```
%(* this program calculates q(8) for one ring network *)
%(* the coefficient values for d and a_d *)
d =[10 12 14 16 18 10];
ad=[11 38 193 1331 7275 40406];

%(* the number of interval for integration *)
interv=5;

%(* the values for R_1 and R_2 *)
r1=[0 0];
r2=[50 50];

%(* set value of α, L, number of user, and m1 for M *)
alpha=4;
L=400;
nuser=8;
m1=[1 2 4];

%(* calculate the grid points of integration *)
delta(1)=(r2(1)-r1(1))/interv;
delta(2)=(r2(2)-r1(2))/interv;
r(1,1)=r1(1)+delta(1)/2;
r(2,1)=r1(2)+delta(2)/2;
for i=2:interv
   r(1,i)=r(1,i-1)+delta(1);
end
for i=2:interv
   r(2,i)=r(2,i-1)+delta(2);
end

%(* calculate the probability density function f_R equation (7) *)
for i=1:interv
   fr(1,i)=2*r(1,i)/(r2(1)^2-r1(1)^2);
end
for i=1:interv
   fr(2,i)=2*r(2,i)/(r2(1)^2-r1(2)^2);
end
```

```
%(* this loop calculate q(8) for all
%value of M=1, 2, and 4, equation (38) *)
for i2=1:3
   m=m1(i2);
   sum=0;
   for a=1:interv
     for b=1:interv
       for c=1:interv
         for f=1:interv
           for g=1:interv
             for h=1:interv
               for k=1:interv
                 for o=1:interv
                     index=[a b c f g h k o];
                     muy=0;

                     %(* calculate value for μ, equation (27) *)
                     for i1=1:nuser-1
                        muy=muy+(r(1,index(1)/r(2,index(i1+1)))^alpha;
                     end
                     muy=1/sqrt(1+muy);

                     %(* calculate value for Pb, equation (25) *)
                     pb=(1-muy)/2

                     %(* calculate the value for PbM, equation (26) *)
                     pbm=0;
                     for cnt=0:m-1
                        pbm=pbm+(comb((m-1+cnt),cnt)*(((1+muy)/2)^cnt));
                     end
                     pbm=(pb^m)*pbm;

                     %(* calculate the value for Pd for d even, equqation (36) *)
                     pd= (2*sqrt(pbm*(1-pbm))).^d;

                     %(* calculate Pe, equation (29) *)
                     pe=ad*pd';
                     if pe>1,pe=1;,end
```

```
            %(* calculate the integration for q(n), equation (38) *)
            inc=(1-pe)^L*fr(1,o)*fr(2,a)*fr(2,b)*fr(2,c)*fr(2,f);
            inc=inc*fr(2,g)*fr(2,h)*fr(2,k);
            sum=sum+inc;
          end
         end
        end
       end
      end
     end
    end
   end
  sum=sum*delta(2)^(nuser-1)*delta(1);
end
```

The source code for the numerical results of Table III is listed below. For ring one, R_1 varies from 0 km to 25 km, and R_2 varies from 0 km to 50 km. This source code is used to calculate q(8). To compute q(2), q(3),...,q(n), the number of users (nuser) is equal to 2, 3,..., n, respectively.

```
%(* this program calculates q(8) for ring one *)
%(* the coefficient values for d and a_d *)
d =[10 12 14 16 18 10];
ad=[11 38 193 1331 7275 40406];

%(* the number of interval for integration *)
interv=5;

%(* the values for R_1 and R_2 *)
r1=[0 0];
r2=[25 50];

%(* set value of α, L, number of user, and m1 for M *)
alpha=4;
L=400;
nuser=8;
m1=[1 2 4];

%(* calculate the grid points of integration *)
delta(1)=(r2(1)-r1(1))/interv;
delta(2)=(r2(2)-r1(2))/interv;
r(1,1)=r1(1)+delta(1)/2;
r(2,1)=r1(2)+delta(2)/2;
for i=2:interv
   r(1,i)=r(1,i-1)+delta(1);
end
for i=2:interv
   r(2,i)=r(2,i-1)+delta(2);
end

%(* calculate the probability density function f_R eq. '.ion (7) *)
for i=1:interv
   fr(1,i)=2*r(1,i)/(r2(1)^2-r1(1)^2);
end
for i=1:interv
   fr(2,i)=2*r(2,i)/(r2(1)^2-r1(2)^2);
end
```

```
%(* this loop calculates q(8) for all
%value of M=1, 2, and 4, equation (38) *)
for i2=1:3
  m=m1(i2);
  sum=0;
  for a=1:interv
   for b=1:interv
    for c=1:interv
     for f=1:interv
      for g=1:interv
       for h=1:interv
        for k=1:interv
         for o=1:interv
           index=[a b c f g h k o];
           muy=0;

           %(* calculate value for μ, equation (27) *)
           for i1=1:nuser-1
             muy=muy+(r(1,index(1)/r(2,index(i1+1)))^alpha;
           end
           muy=1/sqrt(1+muy);

           %(* calculate value for Pb, equation (25) *)
           pb=(1-muy)/2

           %(* calculate the value for PbM, equation (26) *)
           pbm=0;
           for cnt=0:m-1
             pbm=pbm+(comb((m-1+cnt),cnt)*(((1+muy)/2)^cnt));
           end
           pbm=(pb^m)*pbm;

           %(* calculate the value for Pd for d even, equqation (36) *)
           pd=(2*sqrt(pbm*(1-pbm))).^d;

           %(* calculate Pe, equation (29) *)
           pe=ad*pd';
           if pe>1,pe=1;,end
```

```
                    %(* calculate the integration for q(n), equation (38) *)
                    inc=(1-pe)^L*fr(1,o)*fr(2,a)*fr(2,b)*fr(2,c)*fr(2,f);
                    inc= inc*fr(2,g)*fr(2,h)*fr(2,k);
                    sum=sum+inc;
                  end
                end
              end
            end
          end
        end
      end
    end
  sum=sum*delta(2)^(nuser-1)*delta(1);
end
```

The source code for the numerical results of Table IV is listed below. For ring two, R_1 varies from 25 km to 38.5 km, and R_2 varies from 0 km to 50 km. This program is used to calculate q(8). To calculate q(2), q(3),..., q(n), the number of users (nuser) is equal to 2, 3,..., n, respectively.

```
%(* this program will calculate q(8) for ring two *)
%(* the coefficient values for d and a_d *)
d =[10 12 14 16 18 10];
ad=[11 38 193 1331 7275 40406];

%(* the number of interval for integration *)
interv=5;

%(* the values for R_1 and R_2 *)
r1=[25 0];
r2=[38.5 50];

%(* set value of α, L, number of user, and m1 for M *)
alpha=4;
L=400;
nuser=8;
m1=[1 2 4];

%(* calculate the grid points of integration *)
delta(1)=(r2(1)-r1(1))/interv;
delta(2)=(r2(2)-r1(2))/interv;
r(1,1)=r1(1)+delta(1)/2;
r(2,1)=r1(2)+delta(2)/2;
for i=2:interv
   r(1,i)=r(1,i-1)+delta(1);
end
for i=2:interv
   r(2,i)=r(2,i-1)+delta(2);
end

%(* calculate the probability density function f_R equation (7) *)
for i=1:interv
   fr(1,i)=2*r(1,i)/(r2(1)^2-r1(1)^2);
end
for i=1:interv
   fr(2,i)=2*r(2,i)/(r2(1)^2-r1(2)^2);
end
```

```
%(* this loop calculates q(8) for all
%value of M=1, 2, and 4, equation (38) *)
for i2=1:3
  m=m1(i2);
  sum=0;
  for a=1:interv
   for b=1:interv
    for c=1:interv
     for f=1:interv
      for g=1:interv
       for h=1:interv
        for k=1:interv
         for o=1:interv
           index=[a b c f g h k o];
           muy=0;

           %(* calculate value for μ, equation (27) *)
           for i1=1:nuser-1
             muy=muy+(r(1,index(1)/r(2,index(i1+1)))^alpha;
           end
           muy=1/sqrt(1+muy);

           %(* calculate value for P_b, equation (25) *)
           pb=(1-muy)/2

           %(* calculate the value for P_bM, equation (26) *)
           pbm=0;
           for cnt=0:m-1
             pbm=pbm+(comb((m-1+cnt),cnt)*(((1+muy)/2)^cnt));
           end
           pbm=(pb^m)*pbm;

           %(* calculate the value for P_d for d even, equqation (36) *)
           pd=(2*sqrt(pbm*(1-pbm))).^d;

           %(* calculate P_e, equation (29) *)
           pe=ad*pd';
           if pe>1,pe=1;,end
```

```
                    %(* calculate the integration for q(n), equation (38) *)
                    inc=(1-pe)^L*fr(1,o)*fr(2,a)*fr(2,b)*fr(2,c)*fr(2,f);
                    inc=inc*fr(2,g)*fr(2,h)*fr(2,k);
                    sum= sum+inc;
                  end
                end
              end
            end
          end
        end
      end
    end
  sum=sum*delta(2)^(nuser-1)*delta(1);
end
```

The source code for the numerical results of Table V is listed below. For ring three, R_1 varies from 38.5 km to 50 km, and R_2 varies from 0 km to 50 km. This program is used to calculate q(8). To calculate q(2), q(3),..., q(n), the number of users (nuser) is equal to 2, 3,..., n, respectively.

```
%(* this program will calculate q(8) for ring three *)
%(* the coefficient values for d and ad *)
d =[10 12 14 16 18 10];
ad=[11 38 193 1331 7275 40406];

%(* the number of interval for integration *)
interv=5;

%(* the values for R1 and R2 *)
r1=[38.5 0];
r2=[50 50];

%(* set value of α, L, number of user, and m1 for M *)
alpha=4;
L=400;
nuser=8;
m1=[1 2 4];

%(* calculate the grid points of integration *)
delta(1)=(r2(1)-r1(1))/interv;
delta(2)=(r2(2)-r1(2))/interv;
r(1,1)=r1(1)+delta(1)/2;
r(2,1)=r1(2)+delta(2)/2;
for i=2:interv
   r(1,i)=r(1,i-1)+delta(1);
end
for i=2:interv
   r(2,i)=r(2,i-1)+delta(2);
end

%(* calculate the probability density function fR eq  ion (7) *)
for i=1:interv
   fr(1,i)=2*r(1,i)/(r2(1)^2-r1(1)^2);
end
for i=1:interv
   fr(2,i)=2*r(2,i)/(r2(1)^2-r1(2)^2);
end
```

```
%(* this loop calculates q(8) for all
%value of M=1, 2, and 4, equation (38) *)
for i2=1:3
   m=m1(i2);
   sum=0;
   for a=1:interv
     for b=1:interv
       for c=1:interv
         for f=1:interv
           for g=1:interv
             for h=1:interv
               for k=1:interv
                 for o=1:interv
                     index=[a b c f g h k o];
                     muy=0;

                     %(* calculate value for μ, equation (27) *)
                     for i1=1:nuser-1
                         muy=muy+(r(1,index(1)/r(2,index(i1+1)))^alpha;
                     end
                     muy=1/sqrt(1+muy);

                     %(* calculate value for Pb, equation (25) *)
                     pb=(1-muy)/2

                     %(* calculate the value for PbM, equation (26) *)
                     pbm=0;
                     for cnt=0:m-1
                         pbm=pbm+(comb((m-1+cnt),cnt)*(((1+muy)/2)^cnt));
                     end
                     pbm=(pb^m)*pbm;

                     %(* calculate the value for Pd for d even, equqation (36) *)
                     pd=(2*sqrt(pbm*(1-pbm)))^d;

                     %(* calculate Pe, equation (29) *)
                     pe=ad*pd';
                     if pe>1,pe=1;,end
```

```
                %(* calculate the integration for q(n), equation (38) *)
                inc=(1-pe)^L*fr(1,o)*fr(2,a)*fr(2,b)*fr(2,c)*fr(2,f);
                inc=inc*fr(2,g)*fr(2,h)*fr(2,k);
                sum=sum+inc;
              end
            end
          end
        end
      end
    end
   end
  end
 sum=sum*delta(2)^(nuser-1)*delta(1);
end
```

APPENDIX B

Equation (36) is used to calculate q(n) since computing q(n) for $n \geq 6$ from equation (30) is prohibitively time-consuming. The following programs use (30) for P_d to calculate q(n). The following programs are listed for reference purposes. The source codes and the results are listed below. Since q(n) for $n > 6$ takes a long time to calculate, the program is written to calculate only up to q(6).

The number of users (nuser) is initialized to 2, 3,..., 6 to calculate q(2), q(3), ..., q(6), respectively. In the listed program the number of users (nuser) is equal to six. For a one-ring network, R_1 and R_2 vary from 0km to 50km. The source code and its simulation results are listed below and in Table VI, respectively.

```
%(* this program calculates q(6) for one ring network *)
%(* the coefficient values for d and a_d *)
d =[10 12 14 16 18 10];
ad=[11 38 193 1331 7275 40406];

%(* the number of interval for integration *)
interv=5;

%(* the values for R_1 and R_2 *)
r1=[0 0];
r2=[50 50];

%(* set value of α, L, number of user, and m1 for M *)
alpha=4;
L=400;
nuser=8;
m1=[1 2 4];

%(* calculate the grid points of integration *)
delta(1)=(r2(1)-r1(1))/interv;
delta(2)=(r2(2)-r1(2))/interv;
r(1,1)=r1(1)+delta(1)/2;
r(2,1)=r1(2)+delta(2)/2;
for i=2:interv
   r(1,i)=r(1,i-1)+delta(1);
end
for i=2:interv
   r(2,i)=r(2,i-1)+delta(2);
end

%(* calculate the probability density function f_R equation (7) *)
for i=1:interv
   fr(1,i)=2*r(1,i)/(r2(1)^2-r1(1)^2);
end
for i=1:interv
   fr(2,i)=2*r(2,i)/(r2(1)^2-r1(2)^2);
end
```

```
%(* this loop calculate q(6) for all
%value of M=1, 2, and 4, equation (38) *)
for i2=1:3
  m=m1(i2);
  sum=0;
  for a=1:interv
    for b=1:interv
      for c=1:interv
        for f=1:interv
          for g=1:interv
            for o=1:interv
                    index=[a b c f g o];
                    muy=0;

                    %(* calculate value for μ, equation (27) *)
                    for i1=1:nuser-1
                        muy=muy+(r(1,index(1)/r(2,index(i1+1)))^alpha;
                    end
                    muy=1/sqrt(1+muy);

                    %(* calculate value for Pb, equation (25) *)
                    pb=(1-muy)/2

                    %(* calculate the value for PbM, equation (26) *)
                    pbm=0;
                    for cnt=0:m-1
                        pbm=pbm+(comb((m-1+cnt),cnt)*(((1+muy)/2)^cnt));
                    end
                    pbm=(pb^m)*pbm;

                    %(* calculate the value for Pd for d even, equqation (30) *)
                    for q=1:6
                        pd(q)=0.5*comb(d(q),d(q)/2)*pbm^(d(q)/2);
                        pd(q)=pd(q)*(1-pbm)^(d(q)/2);
                        for e=1+d(q)/2:d(q)
                        pd(q)=pd(q)+(comb(d(q),e)*(pbm^e)*(1-pbm)^(d(q)-e));
                        end
                    end
```

```
              %(* calculate P_e, equation (29) *)
              pe=ad*pd';
              if pe>1, pe=1;, end

              %(* calculate the integration for q(n), equation (38) *)
              inc=(1-pe)^L*fr(1,o)*fr(2,a)*fr(2,b)*fr(2,c)*fr(2,f);
              inc=inc*fr(2,g);
              sum=sum+inc;
            end
          end
        end
      end
    end
  end
 sum=sum*delta(2)^(nuser-1)*delta(1);
end
```

Table VI. SIMULATION RESULTS FOR THE PROBABILITY OF CORRECT PACKET CAPTURE FOR ONE-RING NETWORK

	q(1)	q(2)	q(3)	q(4)	q(5)	q(6)
M=1	1	.228	.138	.096	.075	.059
M=2	1	.412	.238	.159	.123	.096
M=4	1	.636	.442	.314	.241	.197

The source code and the simulation results are listed below and in Table VII, respectively. For ring one R_1 is varied from 0 km to 25 km, and R_2 is varied from 0 km to 50 km. This source code is used to calculate q(6). To compute q(2), q(3),...,q(n), the number of users (nuser) is 2,3,...,n respectively.

```
%(* this program calculates q(6) for ring one *)
%(* the coefficient values for d and a_d *)
d =[10 12 14 16 18 10];
ad=[11 38 193 1331 7275 40406];

%(* the number of interval for integration *)
interv=5;

%(* the values for R_1 and R_2 *)
r1=[0 0];
r2=[25 50];

%(* set value of α, L, number of user, and m1 for M *)
alpha=4;
L=400;
nuser=8;
m1=[1 2 4];

%(* calculate the grid points of integration *)
delta(1)=(r2(1)-r1(1))/interv;
delta(2)=(r2(2)-r1(2))/interv;
r(1,1)=r1(1)+delta(1)/2;
r(2,1)=r1(2)+delta(2)/2;
for i=2:interv
   r(1,i)=r(1,i-1)+delta(1);
end
for i=2:interv
   r(2,i)=r(2,i-1)+delta(2);
end

%(* calculate the probability density function f_R equation (7) *)
for i=1:interv
   fr(1,i)=2*r(1,i)/(r2(1)^2-r1(1)^2);
end
for i=1:interv
   fr(2,i)=2*r(2,i)/(r2(1)^2-r1(2)^2);
end
```

```
%(* this loop calculates q(6) for all
%value of M=1, 2, and 4, equation (38) *)
for i2=1:3
  m=m1(i2);
  sum=0;
  for a=1:interv
   for b=1:interv
    for c=1:interv
     for f=1:interv
      for g=1:interv
       for o=1:interv
           index=[a b c f g o];
           muy=0;

           %(* calculate value for μ, equation (27) *)
           for i1=1:nuser-1
             muy=muy+(r(1,index(1)/r(2,index(i1+1)))^alpha;
           end
           muy=1/sqrt(1+muy);

           %(* calculate value for Pb, equation (25) *)
           pb=(1-muy)/2

           %(* calculate the value for PbM, equation (26) *)
           pbm=0;
           for cnt=0:m-1
             pbm=pbm+(comb((m-1+cnt),cnt)*(((1+muy)/2)^cnt));
           end
           pbm=(pb^m)*pbm;

           %(* calculate the value for Pd for d even, equqation (30) *)
           for q=1:6
             pd(q)=0.5*comb(d(q),d(q)/2)*pbm^(d(q)/2);
             pd(q)=pd(q)*(1-pbm)^(d(q)/2);
             for e=1+d(q)/2:d(q)
             pd(q)=pd(q)+(comb(d(q),e)*(pbm^e)*(1-pbm)^(d(q)-e));
             end
           end
```

```
            %(* calculate Pc, equation (29) *)
             pe=ad*pd';
             if pe>1, pe=1;, end

             %(* calculate the integration for q(n), equation (38) *)
             inc=(1-pe)^L*fr(1,o)*fr(2,a)*fr(2,b)*fr(2,c)*fr(2,f);
             inc=inc*fr(2,g);
             sum=sum+inc;
           end
          end
         end
        end
       end
      end
  sum=sum*delta(2)^(nuser-1)*delta(1);
end
```

Table VII. SIMULATION RESULTS FOR THE PROBABILITY OF CORRECT PACKET CAPTURE FOR RING ONE

	q(1)	q(2)	q(3)	q(4)	q(5)	q(6)
M=1	1	.751	.505	.355	.2838	.238
M=2	1	.881	.752	.624	.5169	.437
M=4	1	.916	.836	.761	.688	.617

The source code and the simulation results are listed below and in Table VIII, respectively. For ring two, R_1 is varied from 25 km to 38.5 km, and R_2 is varied from 0 km to 50 km. This program is used to calculate q(6). To calculate q(2), q(3),..., q(n), the number of users (nuser) is 2, 3,..., n respectively.

```
%(* this program will calculate q(6) for ring two *)
%(* the coefficient values for d and a_d *)
d =[10 12 14 16 18 10];
ad=[11 38 193 1331 7275 40406];

%(* the number of interval for integration *)
interv=5;

%(* the values for R_1 and R_2 *)
r1=[25 0];
r2=[38.5 50];

%(* set value of α, L, number of user, and m1 for M *)
alpha=4;
L=400;
nuser=8;
m1=[1 2 4];

%(* calculate the grid points of integration *)
delta(1)=(r2(1)-r1(1))/interv;
delta(2)=(r2(2)-r1(2))/interv;
r(1,1)=r1(1)+delta(1)/2;
r(2,1)=r1(2)+delta(2)/2;
for i=2:interv
   r(1,i)=r(1,i-1)+delta(1);
end
for i=2:interv
   r(2,i)=r(2,i-1)+delta(2);
end

%(* calculate the probability density function f_R equation (7) *)
for i=1:interv
   fr(1,i)=2*r(1,i)/(r2(1)^2-r1(1)^2);
end
for i=1:interv
   fr(2,i)=2*r(2,i)/(r2(1)^2-r1(2)^2);
end
```

```
%(* this loop calculates q(6) for all
%value of M=1, 2, and 4, equation (38) *)
for i2=1:3
  m=m1(i2);
  sum=0;
  for a=1:interv
   for b=1:interv
    for c=1:interv
     for f=1:interv
      for g=1:interv
       for o=1:interv
           index=[a b c f g o];
           muy=0;

           %(* calculate value for μ, equation (27) *)
           for i1=1:nuser-1
             muy=muy+(r(1,index(1)/r(2,index(i1+1)))^alpha;
           end
           muy=1/sqrt(1+muy);

           %(* calculate value for Pb, equation (25) *)
           pb=(1-muy)/2

           %(* calculate the value for PbM, equation (26) *)
           pbm=0;
           for cnt=0:m-1
             pbm=pbm+(comb((m-1+cnt),cnt)*(((1+muy)/2)^cnt));
           end
           pbm=(pb^m)*pbm;

           %(* calculate the value for Pd for d even, equqation (30) *)
           for q=1:6
             pd(q)=0.5*comb(d(q),d(q)/2)*pbm^(d(q)/2);
             pd(q)=pd(q)*(1-pbm)^(d(q)/2);
             for e=1+d(q)/2:d(q)
             pd(q)=pd(q)+(comb(d(q),e)*(pbm^e)*(1-pbm)^(d(q)-e));
             end
           end
```

```
            %(* calculate P_e, equation (29) *)
            pe=ad*pd';
            if pe>1, pe=1;, end

            %(* calculate the integration for q(n), equation (38) *)
            inc=(1-pe)^L*fr(1,o)*fr(2,a)*fr(2,b)*fr(2,c)*fr(2,f);
            inc=inc*fr(2,g);
            sum=sun+inc;
          end
        end
      end
    end
  end
 end
 sum=sum*delta(2)^(nuser-1)*delta(1);
end
```

Table VIII. SIMULATION RESULTS FOR THE PROBABILITY OF CORRECT PACKET CAPTURE FOR RING TWO

	q(1)	q(2)	q(3)	q(4)	q(5)	q(6)
M=1	1	.055	.0018	0	0	0
M=2	1	.33	.1021	.0227	.0052	.0011
M=4	1	.487	.2971	.1534	.065	.0293

The source code and the simulation results are listed below and in Table IX, respectively. For ring three, R_1 is varied from 38.5 km to 50 km, and R_2 is varied from 0 km to 50 km. This program is used to calculate q(6). To calculate q(2), q(3),..., q(n), the number of users (nuser) is 2, 3,..., n respectively.

```
%(* this program will calculate q(6) for ring three *)
%(* the coefficient values for d and a_d *)
d =[10 12 14 16 18 10];
ad=[11 38 193 1331 7275 40406];

%(* the number of interval for integration *)
interv=5;

%(* the values for R_1 and R_2 *)
r1=[38.5 0];
r2=[50 50];

%(* set value of α, L, number of user, and m1 for M *)
alpha=4;
L=400;
nuser=8;
m1=[1 2 4];

%(* calculate the grid points of integration *)
delta(1)=(r2(1)-r1(1))/interv;
delta(2)=(r2(2)-r1(2))/interv;
r(1,1)=r1(1)+delta(1)/2;
r(2,1)=r1(2)+delta(2)/2;
for i=2:interv
   r(1,i)=r(1,i-1)+delta(1);
end
for i=2:interv
   r(2,i)=r(2,i-1)+delta(2);
end

%(* calculate the probability density function f_R equation (7) *)
for i=1:interv
   fr(1,i)=2*r(1,i)/(r2(1)^2-r1(1)^2);
end
for i=1:interv
   fr(2,i)=2*r(2,i)/(r2(1)^2-r1(2)^2);
end
```

```
%(* this loop calculates q(6) for all
%value of M=1, 2, and 4, equation (38) *)
for i2=1:3
  m=m1(i2);
  sum=0;
  for a=1:interv
   for b=1:interv
    for c=1:interv
     for f=1:interv
      for g=1:interv
       for o=1:interv
           index=[a b c f g o];
           muy=0;

           %(* calculate value for μ, equation (27) *)
           for i1=1:nuser-1
             muy=muy+(r(1,index(1)/r(2,index(i1+1)))^alpha;
           end
           muy=1/sqrt(1+muy);

           %(* calculate value for P_b, equation (25) *)
           pb=(1-muy)/2

           %(* calculate the value for P_bM, equation (26) *)
           pbm=0;
           for cnt=0:m-1
             pbm=pbm+(comb((m-1+cnt),cnt)*(((1+muy)/2)^cnt));
           end
           pbm=(pb^m)*pbm;

           %(* calculate the value for P_d for d even, equqation (30) *)
           for q=1:6
             pd(q)=0.5*comb(d(q),d(q)/2)*pbm^(d(q)/2);
             pd(q)=pd(q)*(1-pbm)^(d(q)/2);
             for e=1+d(q)/2:d(q)
             pd(q)=pd(q)+(comb(d(q),e)*(pbm^e)*(1-pbm)^(d(q)-e));
             end
           end
```

```
                %(* calculate P_e, equation (29) *)
                pe=ad*pd';
                if pe>1, pe=1;, end

                %(* calculate the integration for q(n), equation (38) *)
                inc=(1-pe)^L*fr(1,o)*fr(2,a)*fr(2,b)*fr(2,c)*fr(2,f);
                inc=inc*fr(2,g);
                sum=sum+inc;
              end
            end
          end
        end
      end
    end
  sum=sum*delta(2)^(nuser-1)*delta(1);
end
```

Table IX. SIMULATION RESULTS FOR THE PROBABILITY OF CORRECT PACKET CAPTURE FOR RING THREE

	q(1)	q(2)	q(3)	q(4)	q(5)	q(6)
M=1	1	0	0	0	0	0
M=2	1	.0474	0	0	0	0
M=4	1	.2578	.0447	.003	.002	0

Plots of q(n) versus n for one-ring and three-ring systems are shown in Figure 11, 12, 13, and 14.

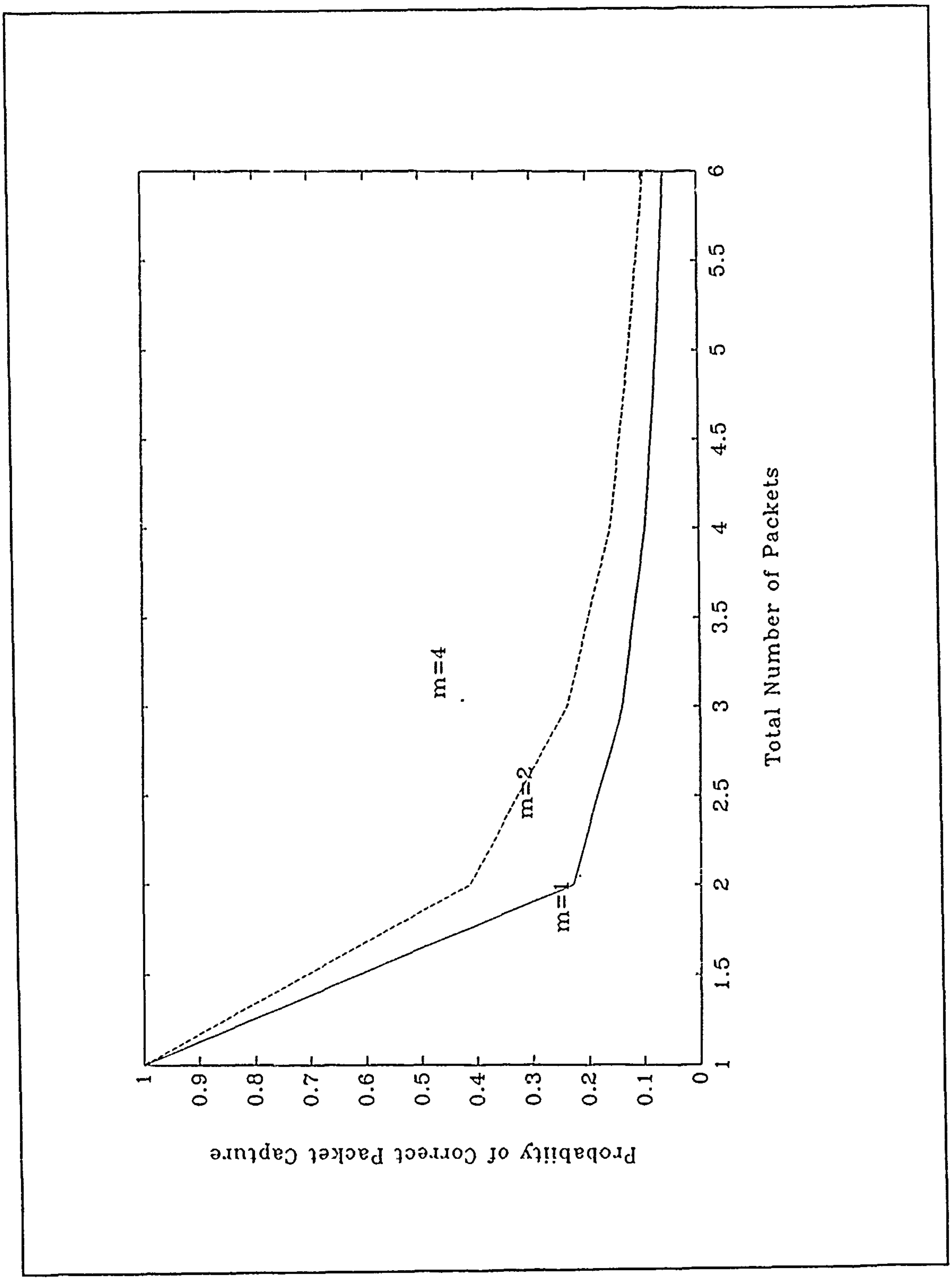

Figure 11. Probability Of Correct Packet Capture For One-Ring Network

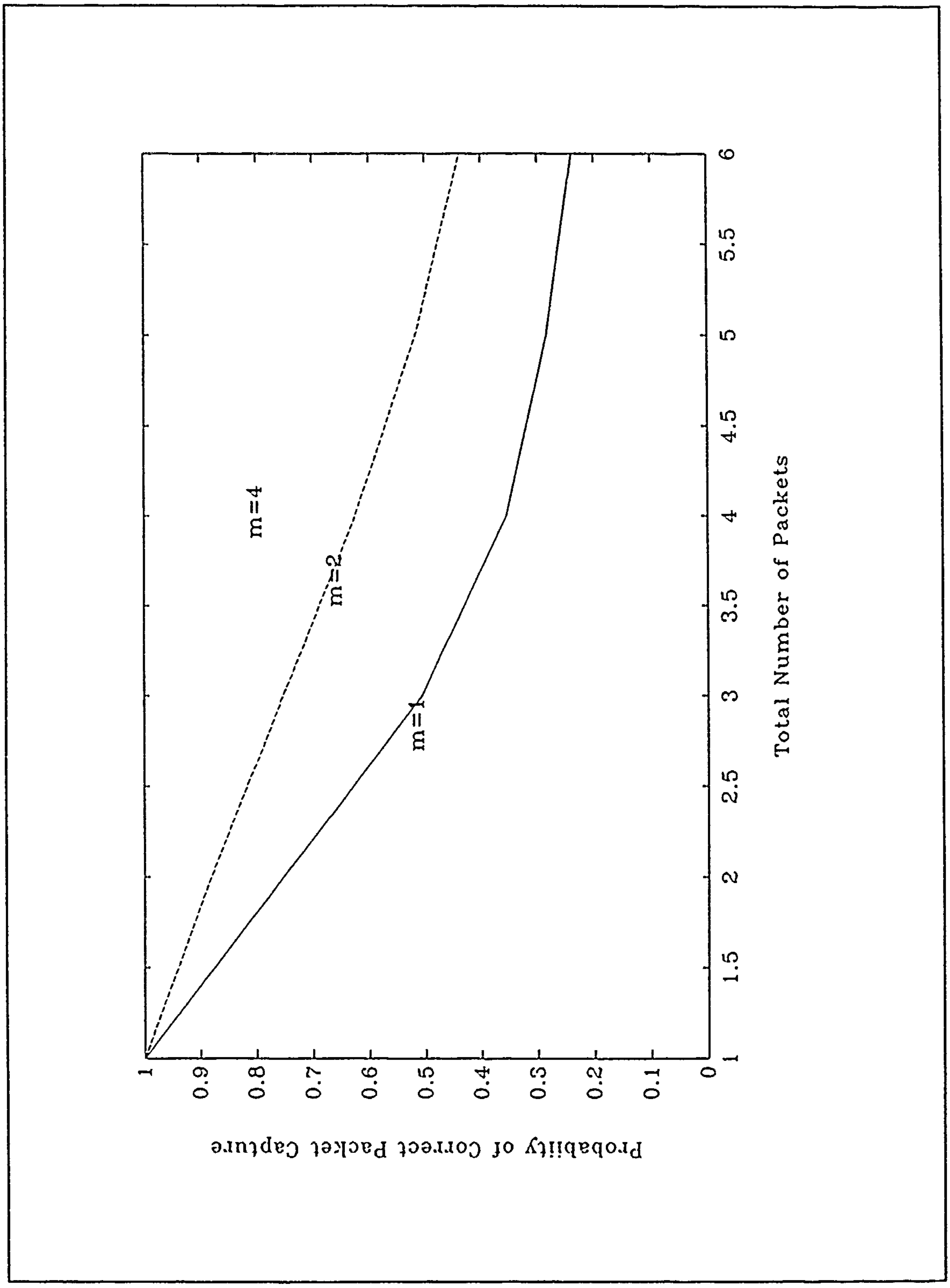

Figure 12. Probability Of Correct Packet Capture For Ring One

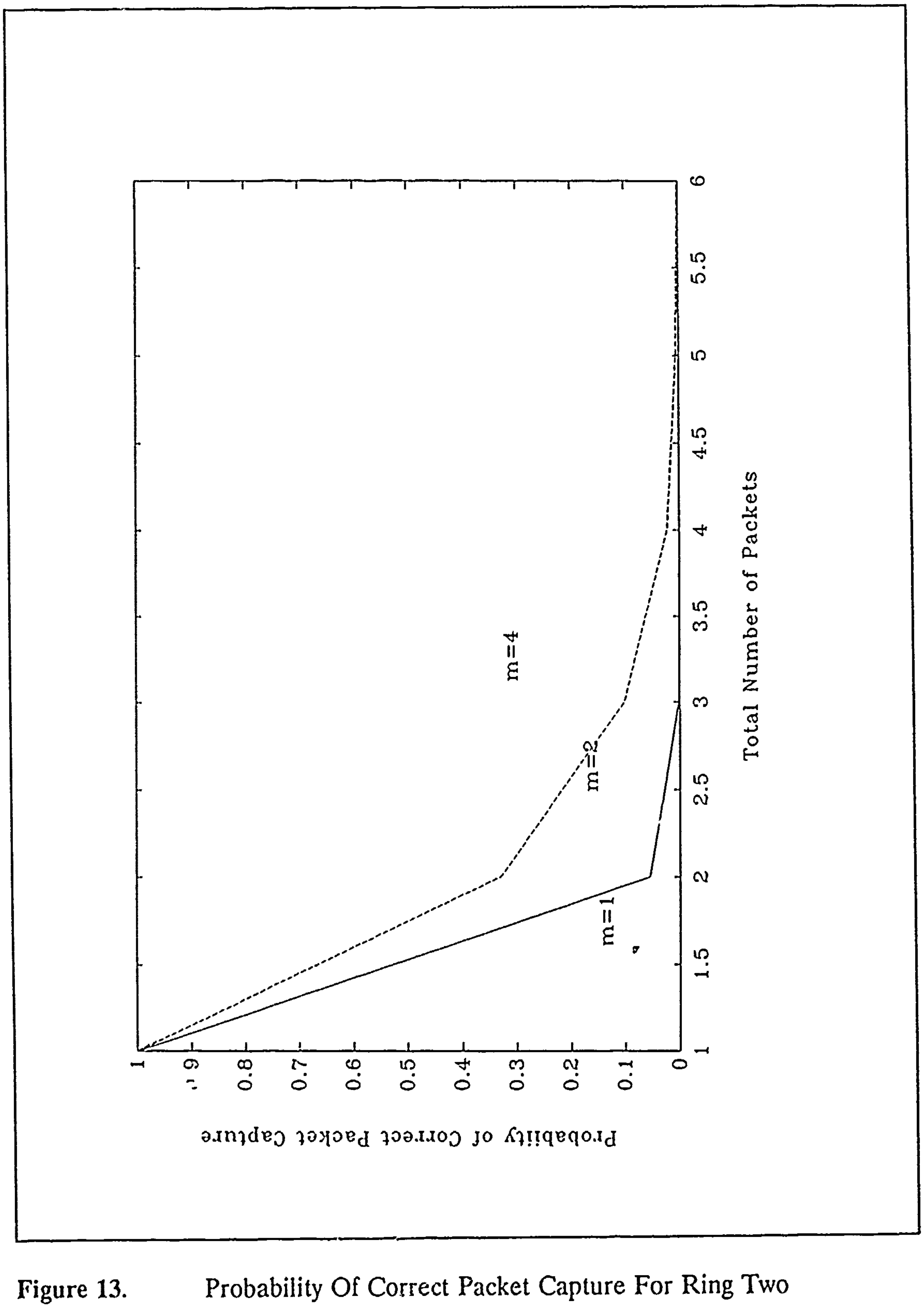

Figure 13. Probability Of Correct Packet Capture For Ring Two

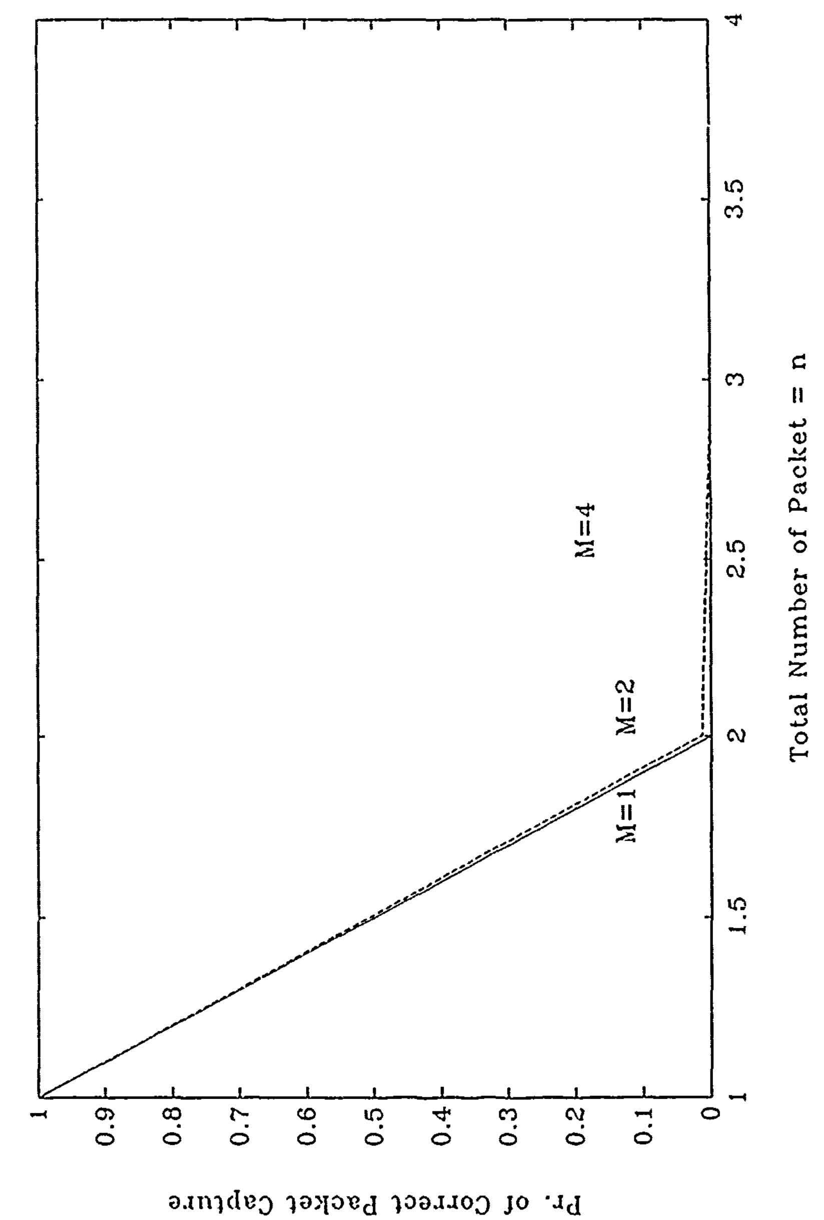

Figure 14. Probability Of Correct Packet Capture For Ring Three

APPENDIX C

```
%(* initialization *)
nuser=15;
p0=0.1;
pr=0.002;
q=[0 1 0.9042 0.8099 0.7197 0.6350 0.5580 0.4901 0.4615 0.4285];

%(* steady state value of q(n) *)
for i=11:nuser+4
   q(i)=0.3852;
end

%(* set N=nuser, calculate π_n,m, it is called pi(n,m) *)
N=nuser;
for n=0:N
  n1=n+1;
  for m=0:N
    m1=m+1;
      if m<n-1,pi(n1,m1)=0;

      elseif m==n-1
        sum1=0;
        for i=0:n
          i1=i+1;
          sum1=sum1+comb(n,i)*pr^i*(1-pr)^(n-i)*q(i1);
        end
        pi(n1,m1)=(1-p0)^(N-n)*sum1;

        elseif m>=n
          sum1=0;
          temp1=comb(N-n,N-m)*(1-p0)^(N-m-1)*p0^(m-n);
          for i=0:n
           i1=i+1
           term=comb(n,i)*(1-pr)^(n-i)*pr;
           term=term*((1-p0)*(1-q(i1+m-n))+(N-m)*p0*q(i1+m-n+1)/(m-n+1));
           sum1=sum1+term;
          end
          pi(n1,m1)=temp1*sum1;
```

```
%(* calculate πn*, it is called pis(n) *)
for n=1:N
  n1=n+1;
  sum1=0;
  for i=0:n-1
    i1=i+1;
    sum1=sum1+(pis(i1)*pi(i1,n1-1));
  end
  temp=pis(n1-1)-sum1;
  pis(n1)=(1/pi(n1,n1-1))*temp;
  if pis(n1)<0,pis(n1)=0;,end
end

%(* calculate πn, it is called pin *)
den=sum(pis);
pin=pis/den;
%(* calculate fn *)
for n=0:N
  n1=n+1;
  temp=0;
  total=0;
  for k=0:N-n
    k1=k+1;
    sum1=0;
    for i=0:n
      sum1=sum1+(comb(n,i)*pr^i*(1-pr)^(n-i)*q(i+k1);
    end
    temp=comb(N-n,N-n-k)*p0^k(1-p0)^(N-n-k);
    term=sum1*temp;
    total=total+term;
  end
  f(n1)=total;
end
%(* calculate s *)
sum1=0;
for i=0:N
  i1=i+1;
  sum1=sum1+(pin(i1)*f(i1));
  s(i1)=sum1;
end
```

```
%(* calculate the Comb(n,m) *)
function y=comb(n,m)
y=xfact(n+1)/(xfact(m+1)*xfact(n-m+!));

%(* calculate the factorial, it is call fact(n) *)
function y=fact(x)
if x==0
  y=1;
else
  y=1;
  for i=1:x
     y=y*i;
  end
end

%(* the variable xfact *)
x=0:52
for i=1:53
   xfact(i)=fact(x(i));
end
global xfact
```

REFERENCES

1. Ha, T. T., *Digital Satellite Communications*, 2d ed., pp. 362-378, McGraw-Hill Publishing Co.,1989.

2. McCartin, J. T., and Ha T. T., "Ring Model For Local Radio Communications," Globecom' 89 Dallas, TX, pp. 20.2.1-20.2.5, 1989.

3. Robertson, R. C., and Ha, T. T., "A Model For Local/Mobile Radio Communications with Correct Packet Capture", *IEEE Trans. Comm.*, forthcoming.

4. Proakis, J. G., *Digital Communications*, 2d ed., New York: McGraw Hill, 1989.

5. Shu Lin, and Costello, J. D., Jr., *Error Control Coding: Fundamentals and Applications*, pp. 322-326, New Jersey: Prentice Hall, Inc., 1983.

6. Namislo, C., "Aloha Networks In Which Collisions Do Not Necessarily Destroy All Packets Involved," *Proc. Mediterarranean Electrotech. Conf.*, pp. A2.09, Athens, Greece, May 1983.

7. Habbab,I. M.I., Kavehrad, M., and Sundberg, C. W., "Aloha With Capture Over Slow and Fast Fading Radio Channels with Coding and Diversity, " *J. Sel. Areas Communications*, vol. 7, no. 1, pp. 79-88, June 1989.

8. Namislo, C., "Analysis of Mobile Radio Slotted ALOHA Networks," *IEEE J. Sel. Areas in Comm.*, vol. SAC-2, no.4, pp. 583-586, July 1984.

INITIAL DISTRIBUTION LIST

		No. of Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, California 93943-5000	2
3.	Department Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5002	1
4.	Professor Tri T. Ha, Code EC/Ha Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93942-5000	3
5.	Professor R. Clark Robertson, Code EC/Rc Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	2
6.	AFCPMC/DPCMW Attn: Barbara A. Hayman Randolph AFB, Texas 78150-6421	1
7.	Kim J. Tran Electro-optical Laboratory, Code ECEL Naval Postgraduate School Monterey, California 93943-5000	2